



Universidade do Minho
Departamento de Sistemas de Informação

Manuel Fernando da Silva Rodrigues

**Proposta de uma Framework para
Desenvolvimento de Sistemas Tutores
Inteligentes**

Tese de Mestrado
Mestrado em Sistemas de Informação

Trabalho efectuado sob a orientação de:
Professor Doutor Paulo Novais
Professor Doutor Manuel Filipe Santos

Junho de 2007

Agradecimentos

Aos meus pais, que sempre me incentivaram a estudar.

Aos meus orientadores, Professor Doutor Paulo Novais e Professor Doutor Manuel Filipe Santos que sempre me apoiaram com as suas críticas e sugestões, e sem os quais não teria conseguido realizar este trabalho.

À Tatiana e à Carolina, as duas mulheres da minha vida, que várias vezes me tiveram ausente, mas sempre compreenderam e me incentivaram.

Resumo

Este trabalho reflecte um levantamento sobre a área dos Sistemas Tutores Inteligentes, focando a sua evolução e características típicas. As arquitecturas propostas por vários investigadores são referenciadas, e a arquitectura *clássica* é detalhada em cada um dos seus módulos constituintes. É feita uma breve análise ao desenvolvimento de Sistemas Tutores Inteligentes, analisando as debilidades e pontos fortes de cada sistema. Tendo por base o estudo efectuado, é proposta uma *Framework* de Desenvolvimento de Sistemas Tutores Inteligentes que procura precisamente colmatar essas debilidades, mantendo as boas práticas de desenvolvimento, incorporando as características desejáveis que foram identificadas, no sentido de “apontar o caminho” para o que se designou como uma nova geração de Sistemas Tutores Inteligentes.

Abstract

In this work a survey in the field of Intelligent Tutoring Systems is done, focusing their evolution and main typical characteristics. Some of the Intelligent Tutoring Systems architectures proposed by researchers in the field are briefly enumerated and explained, and the classical architecture is detailed in each of its component modules. The development of Intelligent Tutoring Systems, together with some of the systems developed so far, is used to identify the weakness and strong issues of each system. A development Framework for Intelligent Tutoring Systems is proposed, which tries to address those weaknesses and incorporate the desirable characteristics that were identified, in order to “show the way” to what was named “The New Generation of Intelligent Tutoring Systems”.

Conteúdo

FIGURAS	VII
TABELAS	VIII
LISTA DE ABREVIATURAS.....	IX
INTRODUÇÃO	1
1.1. Objectivos	3
1.2. Organização da Tese	4
PARTE 1 - SISTEMAS TUTORES INTELIGENTES.....	5
1.3. Tutores Inteligentes.....	5
1.4. Evolução dos Sistemas de Ensino por Computador	8
1.5. IAC – Instrução Assistida por Computador	9
1.6. IIAC – Instrução Inteligente Assistida por Computador	14
1.7. Características dos Sistemas Tutores Inteligentes.....	20
1.8. Conclusões.....	22
2. Architecturas de Sistemas Tutores Inteligentes.....	23
2.1. Estrutura Típica de um Sistema Tutor Inteligente	23
2.2. Architecturas de Sistemas Tutores Inteligentes	24
2.3. Módulo do Domínio.....	27
2.4. Módulo Tutor	28
2.5. Interface	34
2.6. Módulo Aluno	35
2.7. Conclusões.....	42
DESENVOLVIMENTO DE SISTEMAS TUTORES INTELIGENTES	43
2.8. Análise	43
2.9. Conclusões.....	56

PARTE 2 – NOVA GERAÇÃO DE SISTEMAS TUTORES INTELIGENTES .. 57

FD-STI – FRAMEWORK DE DESENVOLVIMENTO DE SISTEMAS TUTORES INTELIGENTES 57

2.10.	Reutilização.....	71
2.11.	Normalização	75
2.12.	Ontologias	78
2.13.	Agentes de Software.....	79
2.14.	Data Mining	86
2.15.	Hipermédia Adaptativa	90
2.16.	Sistemas de Recomendação e Personalização	97
2.17.	Raciocínio Baseado em Casos.....	102
2.18.	Teorias de Aprendizagem.....	107
2.18.1.	Teoria da Instrução de Gagné	109
2.18.2.	Teoria do Equilíbrio de Piaget	110
2.18.3.	Teoria Sociocultural de Vigotsky.....	110
2.18.4.	Teoria de Bruner	111
2.18.5.	Teoria da Aprendizagem Significativa de Ausubel	112
2.18.6.	Inteligências Múltiplas de Gardner.....	112
2.18.7.	Teoria UNO de Perkins	114
2.19.	Base de Conhecimento	118
2.20.	Base de Dados de Modelos Pedagógicos.....	119
2.21.	Conclusões.....	120

CONCLUSÕES 121

REFERENCIAS..... 124

Figuras

<i>Figura 1 - Domínio multidisciplinar dos STI</i>	<i>6</i>
<i>Figura 2 - Estrutura típica de um STI.....</i>	<i>24</i>
<i>Figura 3 - Arquitectura de um STI, adaptado de [Burns, Capps, 88].....</i>	<i>26</i>
<i>Figura 4 - Arquitectura de um STI proposta por McTaggart, adaptado de [Mctaggart, 01]</i>	<i>27</i>
<i>Figura 5 – Arquitectura de um STI segundo Self, adaptado de [Self, 99].....</i>	<i>27</i>
<i>Figura 6 – Modelo de sobreposição Figura 7 – Modelo Diferencial.....</i>	<i>37</i>
<i>Figura 8 – Modelo de Perturbação.....</i>	<i>37</i>
<i>Figura 9 – Arquitectura genérica do tutor SPARSE-IT, adaptado de [Faria, 02].....</i>	<i>49</i>
<i>Figura 10 – Framework de Desenvolvimento de STI com reutilização – adaptado de [El-Sheikh et all, 98].</i>	<i>63</i>
<i>Figura 11 – FD-STI – Framework de Desenvolvimento de Sistemas Tutores Inteligentes.....</i>	<i>69</i>
<i>Figura 12 - Agente a interactivar com o meio</i>	<i>80</i>
<i>Figura 13 - Ciclo clássico dos sistemas adaptativos adaptado de [Brusilovsky, 96]</i>	<i>91</i>
<i>Figura 14 - Taxonomia das Tecnologias de Hipermedia Adaptativa, adaptado de [Brusilovsky, 01].</i>	<i>96</i>
<i>Figura 15 - Arquitectura para personalização off-line, adaptado de [Reategui, Lorenzatti, 05].....</i>	<i>101</i>
<i>Figura 16 - Arquitectura para personalização on-line, adaptado de [Santos et all, 06].....</i>	<i>101</i>
<i>Figura 17 - Modelo ilustrativo do raciocínio baseado em casos, baseado em [Riesbeck, Schank, 91]... </i>	<i>106</i>

Tabelas

<i>Tabela 1 – Categorias e Evolução dos IAC.....</i>	<i>13</i>
<i>Tabela 2 – Princípios da Teoria ACT* e implicações para os STI.</i>	<i>17</i>
<i>Tabela 3 - Teorias Pedagógicas e o Processo Ensino-Aprendizagem, adaptado de [Mizukami, Nicoletti, 86]</i>	<i>30</i>
<i>Tabela 4 - Características de STI desenvolvidos até à década de 90 – adaptado de Kinshuck, 1996</i>	<i>34</i>
<i>Tabela 5 – Síntese de STI, adaptado de Park et al, 1987</i>	<i>52</i>
<i>Tabela 6 - Matriz Problemas/contribuições.....</i>	<i>70</i>
<i>Tabela 7 – Teoria de Piaget vs Teoria de Vigotsky.....</i>	<i>111</i>

Lista de Abreviaturas

STI – Sistema Tutor Inteligente

IA – Inteligência Artificial

WWW – World Wide Web

IAC – Instrução Assistida por Computador

IIAC – Instrução Inteligente Assistida por Computador

FD-STI – Framework de Desenvolvimento de Sistemas Tutores Inteligentes

IP – Instrução Programada

ITS – Intelligent Tutoring System

ACT* – Adaptative Control of Thought

IHM – Interacção Homem Máquina

SMA – Sistema Multiagente

IAD – Inteligência Artificial Distribuída

HA – Hipermédia Adaptativa

RBC – Raciocínio Baseado em Casos

SR – Sistemas de Recomendação

BC – Base de Conhecimento

Introdução

Usar a tecnologia com o objectivo de auxiliar no processo ensino-aprendizagem sempre foi desejado por todos os agentes envolvidos. De facto, as tecnologias disponíveis com o objectivo de auxiliar no ensino são numerosas. Vários investigadores esforçam-se com o intuito de fazer convergir todas essas tecnologias. Após o intenso desenvolvimento do comércio electrónico, com o “boom” da sociedade de informação, a área da educação à distância será certamente um dos focos do momento, se bem que desde há muito tempo que tem suscitado grande interesse e investigação, existindo apenas limitações devido à indisponibilidade da tecnologia necessária à implementação de sistemas de ensino, mais concretamente Sistemas Tutores Inteligentes (STI). Poderemos pois considerar um STI como um programa dotado de inteligência, recorrendo a técnicas de Inteligência Artificial (IA), que consegue iniciar/desenvolver/interferir e avaliar o processo ensino-aprendizagem, definindo o “que ensinar”, “quando ensinar”, e “como ensinar”.

Diversas instituições de ensino e empresas usam a World Wide Web (WWW) para fornecer cursos à distância dos mais variados tipos e para disponibilizar conteúdos. Os ambientes educativos na WWW procuram proporcionar toda a infra-estrutura necessária ao bom funcionamento de tais acções. Recursos como “chats”, fóruns de discussão, listas de e-mails, proporcionam interacção síncrona e assíncrona através de um interface cuidado, graças ao estado actual da tecnologia. Recentemente, a disponibilidade de plataformas de e-learning freeware, tais como o Moodle, e o Joomla por exemplo, têm vindo a massificar o uso de sistemas deste tipo pelas diversas organizações. De facto, praticamente todas as universidades dispõem actualmente duma plataforma do género, inclusivé as escolas secundárias caminham nesse sentido, existindo muitas já com sistemas em funcionamento. No entanto, com o uso de tais sistemas, continuamos a ter um rácio professor/aluno elevado, à semelhança de uma sala de aula tradicional, não existindo ainda a possibilidade de um tutor individualizado. Apesar de se perderem algumas das características de interacção presencial entre aluno /professor, adquirir-se-ão novas características que podem trazer novas funcionalidades e benefícios ao processo de ensino, como por exemplo a disponibilidade espacial e temporal 24/24 horas que tais sistemas proporcionam.

É pois, de certa forma lógico, que o próximo passo seja a disponibilização de sistemas capazes de efectuar a tutoria aos diferentes utilizadores, de uma forma efectiva e individualizada. Tais sistemas têm sido desde algumas décadas objecto de estudo, com o intuito

de proporcionar as características de um tutor humano aos utilizadores daquilo que se designa como STI.

O desenvolvimento de um STI é uma tarefa complexa, dada a natureza do próprio sistema e de todas as variáveis e condicionantes envolvidas. É necessário considerar características relativas ao domínio em questão (os conteúdos), relativas ao comportamento observável do aluno e relativas ao conjunto de estratégias que serão utilizadas para actuar como “tutor” na busca de um ensino personalizado.

Um aspecto importante para possibilitar a construção do conhecimento de cada aluno é perceber o aluno como um ser individual com características próprias. Deste ponto surge a necessidade de investigar formas de modelar o aluno e de proporcionar interacções específicas às suas necessidades.

A sistematização do conhecimento pedagógico a ser colocado no tutor consiste numa das tarefas mais complexas de todo o sistema, relativamente à sua modelação e implementação. A necessidade de incluir mecanismos de apoio à aprendizagem, tais como estratégias e táticas de ensino, baseadas nas utilizadas por professores em sala de aula, é um dos grandes desafios nesta área. A selecção do método pedagógico que se julga mais adequado como resultado da análise do perfil do aluno é um factor extremamente influente na hora de obter bons resultados nas diferentes interacções, sendo portanto de extrema necessidade encontrar técnicas que determinem qual o método potencialmente mais indicado, disponível no sistema (ou seja, qual o que se julga que melhor se adapta às necessidades do aluno), para satisfazer as suas necessidades e desejos, tendo em conta o desempenho anterior e outras características intrínsecas como sejam preferências do aluno e conteúdos a utilizar. Outro aspecto a ter em conta é a necessidade de avaliar constantemente a aprendizagem do aluno em relação ao domínio em estudo, visando uma melhor orientação no sistema. A adaptação e personalização de sistemas aos seus utilizadores pretendem atingir os objectivos referidos acima.

Torna-se pois pertinente e necessária uma reflexão sobre o desenvolvimento de STI, que tenha em conta os aspectos considerados mais relevantes no desenvolvimento de tais sistemas. Um dos aspectos a considerar é o envolvimento das ciências da educação e da psicologia no desenvolvimento de STI. O facto de muitos sistemas serem desenvolvidos apenas por investigadores ligados à área de ciências da computação leva a que surjam muitas vezes potenciais problemas, pois o processo de ensino tem características muito próprias, desde logo

porque lida com pessoas. Este facto é referido muitas vezes como factor de insucesso por vários investigadores.

A inclusão de estilos e teorias de aprendizagem na proposta de *Framework* desta dissertação torna-se pois essencial, dado o domínio em que se insere. Se pensamos num sistema destinado ao ensino, não podemos dissociar desse sistema o estudo da forma como os indivíduos aprendem, bem como as diferentes teorias relativas a essa aprendizagem. Já no que concerne às tecnologias da Informática, são referenciadas na *Framework* proposta algumas que se julga serem importantes no desenvolvimento de STI, e nas quais se têm concentrado as pesquisas mais recentes.

A *Framework* apresentada torna-se bastante ambiciosa, provavelmente demais até, ao sugerir de certa forma que um STI poderia ser completamente independente do domínio em que se insere (ao referir que a base de conhecimento estaria fora do próprio sistema). Percebe-se esta “demasiada ambição” se pensarmos que um professor deveria ser capaz de ensinar qualquer domínio independentemente da sua formação...

Ao colocar também uma base de modelos pedagógicos fora do próprio STI, está-se também, e mais uma vez, a reiterar a importância das teorias da aprendizagem, da pedagogia, num tal sistema, tornando-o dinâmico na forma como ensina, ou seja, o professor ideal que muda de estratégias, personalidade, conhecimentos e forma de actuação perante situações e alunos diferentes.

1.1. Objectivos

É objectivo deste trabalho proceder ao levantamento, a um “*survey*” do desenvolvimento de STI, das arquitecturas propostas e das tecnologias utilizadas, tentando posteriormente efectuar a sua síntese numa *Framework* para o desenvolvimento de Sistemas Tutores Inteligentes de nova geração. Para tal, será efectuado um estudo sobre a área de STI, fazendo inicialmente um enquadramento histórico e teórico, desde o surgimento desta área de investigação até ao que consideramos hoje ser um STI. A arquitectura tradicional tripartida de um STI será abordada, bem como outras arquitecturas propostas por diferentes investigadores. Alguns STI relevantes serão referenciados.

Pretende-se com a *Framework* proposta, sugerir formas de colmatar algumas deficiências apontadas por investigadores, nomeadamente o pouco envolvimento das ciências da educação e da psicologia no desenvolvimento de STI. Algumas áreas do vasto campo das ciências da educação são também referenciadas por se julgar que podem ter um papel fundamental neste processo.

É pois, objectivo deste estudo fornecer uma contribuição que se pensa válida para o desenvolvimento de STI, sob a forma de uma proposta de *Framework* para o desenvolvimento de STI de nova geração.

1.2. Organização da Tese

Este documento está organizado em duas partes. Na primeira parte é apresentado o estado da arte relativamente ao desenvolvimento de STI, começando pelo ponto um, onde é feita uma abordagem dos STI, com o intuito de um enquadramento teórico da área, referindo-se a origem e evolução. As características de um STI são também referenciadas. No ponto dois são abordadas as arquitecturas de um STI, sendo referida a sua estrutura típica, bem como diferentes arquitecturas propostas por vários investigadores. Partindo da arquitectura dita clássica de um STI, são referidos os quatro modelos dos quatro módulos principais, Modelo do Domínio, Modelo Tutor, Interface e Modelo do Aluno. No ponto três, alguns STI desenvolvidos são abordados, relativamente às suas características, pontos fortes e limitações. Finalmente no ponto quatro, são abordados os problemas e limitações de um STI, propondo-se novas funcionalidades para STI ditos de nova geração. É apresentada uma *Framework* para o desenvolvimento de STI, procurando-se justificar a sua necessidade e abordando os tópicos que a constituem. No capítulo cinco são apresentadas as conclusões deste trabalho.

Parte 1 - Sistemas Tutores Inteligentes

Nesta parte da dissertação irá ser abordada a evolução dos sistemas de ensino por computador desde as suas origens até aos dias de hoje. São referidas as suas características e a sua estrutura típica. Algumas arquitecturas propostas por investigadores são referenciadas, sendo de seguida abordados os três módulos principais da arquitectura dita clássica: Modelo Tutor, Modelo Domínio e Modelo Aluno, bem como o Modelo de Interface. Alguns sistemas desenvolvidos são brevemente referidos. Algumas definições para Sistemas Tutores Inteligentes são enunciadas.

1.3. Tutores Inteligentes

Desde os primórdios do ensino, fornecer conhecimento a todos os que o procuram, de modo individualizado e sempre que necessário (a qualquer hora e em qualquer lugar), quando pedido, idealmente mesmo quando esse suporte não é solicitado mas a sua necessidade existe, tem merecido a atenção dos investigadores que procuram utilizar as tecnologias de informação na área da educação.

Existem diversos estudos que demonstram que a tutoria individual (um para um) é mais eficiente que outros modos de instrução, como por exemplo o tradicional modo de instrução em sala de aula. O grande desafio reside na transposição para os computadores da experiência, das capacidades (“skills”) e do modo de actuação do tutor humano, eliminando dessa forma as restrições temporais, de espaço, socio-económicas e ambientais que caracterizam a tutoria humana. Os Sistemas Tutores Inteligentes (STI) surgem assim como uma forma de contornar estas questões, e têm sido objecto de muitos trabalhos de investigação, [Cohen et all, 82], [Bloom, 84], [AutoTutor], [Circle], [Circsim].

O objectivo central é proporcionar os benefícios da tutoria de um para um, de forma automática, reduzindo os custos associados, tendo em vista a natureza multidisciplinar dos STI como mostra a Figura 1.

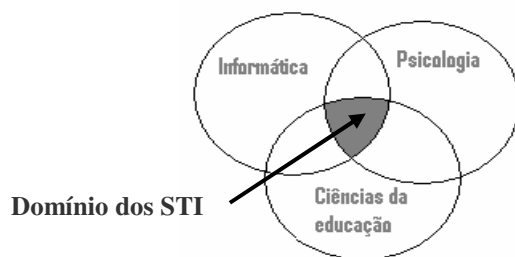


Figura 1 - Domínio multidisciplinar dos STI

O processo de aprendizagem é um processo activo, de aquisição de conhecimentos, em que o aluno cria uma interpretação pessoal da informação que lhe é disponibilizada. A assimilação de conhecimento é realizada através da integração do novo conhecimento com o conhecimento já existente, através de novas inferências. Estas inferências permitem construir relações entre conceitos ou desenvolver o conhecimento adquirido previamente. Os estudantes não recebem a aprendizagem passivamente, devem construí-la. Como “a essência do conhecimento é a estrutura”, o conhecimento humano pode ser organizado em esquemas, e o novo conhecimento faz-se a partir da interpretação desses esquemas, [Kintsch, 93], [Anderson, 84].

Em contextos em que os alunos têm um reduzido conhecimento do domínio em estudo, estas inferências não são geradas automaticamente. O papel do agente responsável pela orientação do ensino (e.g. um tutor), consiste em utilizar os métodos pedagógicos que julga mais adequados, que levem o aluno a construir estas inferências. Esses métodos/táticas podem também ser organizados com o objectivo de melhorar os aspectos que contribuem para a eficácia da instrução, nomeadamente a motivação do aluno, a sua atenção e empenho, entre outros, [Gagné et al, 92].

Para que um tutor humano alcance um bom desempenho é necessário que este possua perícia, tanto no domínio de ensino, como em áreas como a pedagogia e a comunicação. Adicionalmente, como já foi referido, a tutoria de um para um é um método que facilita a aprendizagem activa ¹no estudante e, apesar de existirem estudos que confirmam este facto, nem sempre é o melhor método comparado com a aprendizagem em grupo, pois depende fortemente da experiência e capacidades do tutor. A escassez de tutores humanos que satisfaçam todos estes requisitos fornece perspectivas bastante animadoras para o uso de sistemas de ensino por computador cujo desempenho se aproxime do apresentado por esses tutores humanos, ou melhor, cujo desempenho se aproxime do tutor ideal, ou seja, aquele que consegue transmitir os conteúdos em estudo, da forma mais adequada e que mais rapidamente leve os alunos a adquirir

¹ Aprendizagem activa é uma forma de instrução que procura envolver o aluno no processo ensino-aprendizagem, é muitas vezes associada ao conceito “aprender fazendo”. Bruner sugere que os alunos que se envolvem activamente com o trabalho que realizam têm mais probabilidades de recordar a informação e aplica-la posteriormente a diferentes contextos.

as competências que se julgam necessárias, [Bruner, 90], [Khuwaja, 94], [Abbas, 98], [Bloom, 84], [Cohen et al, 82].

Não existindo tutores humanos suficientes com tão alta experiência, a necessidade de mais e melhores STI, eficientes no desempenho da sua função, abre excelentes perspectivas para esta área de investigação.

Os tutores humanos requerem dois tipos de experiência:

- No domínio de ensino – pois devem transmitir conhecimentos sobre uma determinada área, devendo idealmente possuir um conhecimento aprofundado sobre os conteúdos a abordar;
- No domínio da tutoria – pois devem transmitir esses conhecimentos da forma mais eficaz possível.

Englobando três aspectos fundamentais:

- Diagnosticar os problemas do aluno;
- Planear a retro-alimentação com novos conhecimentos;
- Comunicar com os alunos.

O que gera algumas questões:

- Como usa o tutor humano os modelos mentais do domínio para solucionar as concepções erradas e ajudar o estudante a construir modelos mentais correctos com base científica?
- Que concepções epistemológicas sobressaem nas suas práticas educativas?
- Como relacionam as questões epistemológicas sobre a natureza do conhecimento a ensinar e como transmitem esse conhecimento?

Os sistemas interactivos do tipo STI devem proporcionar um modo de aprendizagem adaptável de acordo com os conhecimentos prévios e a capacidade de evolução de cada aluno (ritmo de aprendizagem). Por outro lado, cada aluno deveria poder escolher as características do método aplicado pelo tutor de acordo com as suas preferências, de entre os diferentes métodos que utiliza, orientador, socrático ou outros. E, se desejar, deveria poder trocar de acordo com os

seus próprios requisitos, pois desta forma a aprendizagem será vista pelo aluno como um processo mais do seu agrado, logo com maiores probabilidades de sucesso, [Khuwaja, 94].

1.4. Evolução dos Sistemas de Ensino por Computador

Desde sempre, a tutoria individual tem sido considerada pela psicologia educacional o melhor método de aprendizagem para os humanos. É um dado adquirido que um ambiente de ensino personalizado com um professor e com um aluno é muito mais efectivo que o ambiente tradicional da sala de aula, um professor com vários alunos, principalmente devido ao rácio professor/aluno que limita o tempo disponível do professor para cada um dos seus alunos. Os STI são uma potencial solução para este problema, embora as investigações actuais ainda estejam longe de atingir este objectivo e caminhem no sentido de enriquecer a equação do processo ensino/aprendizagem com um novo membro, e não no sentido de substituir o professor/tutor humano, [Cohen et al, 82], [Bloom, 84].

Os STI foram alvos onde os investigadores de Inteligência Artificial (IA) aplicaram várias teorias visando experimentá-las e comprovar a sua validade. No final da década de cinquenta e início dos anos sessenta, a IA era considerada uma área muito promissora, com potencialidades praticamente ilimitadas. Acreditava-se que em poucos anos se conseguiriam obter computadores realmente inteligentes, capazes de pensar e actuar como o ser humano. Muitos defendiam com entusiasmo que para ultrapassar este obstáculo bastaria apenas construir computadores com maior capacidade, mais rápidos. Parecia razoável assumir que, uma vez construídas essas máquinas, com capacidades de raciocínio, estas poderiam assumir qualquer tarefa associada com o pensamento humano, nomeadamente o ensino/instrução. Esses objectivos ainda não foram atingidos, e foram de certa forma redefinidos. No entanto a aplicação das técnicas de IA em diferentes tipos de sistemas originou uma evolução interessante.

1.5. IAC – Instrução Assistida por Computador

Os primeiros sistemas informáticos de suporte ao ensino centravam-se na instrução, não tendo em conta as questões relativas à aprendizagem. Eram chamados de sistemas de Instrução Assistida por Computador (IAC) e tinham por objectivo apresentar os conteúdos a serem leccionados ao aluno, de uma forma sequencial pré-determinada, sem ter em consideração os diferentes tipos de alunos que utilizariam o sistema, bem como outras condicionantes.

Os IAC surgiram nos anos cinquenta, eram constituídos por unidades de ensino que apresentavam ao aluno conteúdos sob a forma de janelas. Estas janelas continham informação sob a forma de textos ou questões (resposta do tipo verdadeiro/falso ou de escolha múltipla). O conhecimento acerca do domínio em estudo estava codificado como parte do texto, não permitindo ao sistema raciocinar acerca do conhecimento do domínio. Algumas das janelas continham testes de avaliação que o sistema apresentava ao aluno para verificar se os conhecimentos tinham sido assimilados. A avaliação das respostas do aluno permitia ao sistema seleccionar a próxima janela a apresentar. Esta avaliação limitava-se a determinar se as respostas estavam correctas ou erradas e a escolher a próxima janela de acordo com a resposta fornecida, não existindo nenhum outro tipo de critério para definir a próxima acção. O percurso de aprendizagem que o aluno podia seguir, estava determinado à partida, embebido no próprio sistema.

Por essa altura, Skinner influenciou a Instrução Programada ²(IP) baseando a sua teoria na ideia que o processo de ensino/aprendizagem é uma função da mudança de comportamento. Mudanças no comportamento são o resultado de reacções individuais (respostas) a eventos (estímulos) que acontecem (num dado ambiente particular). Uma resposta produz uma acção. Quando um padrão específico de estímulo - resposta é verificado, o indivíduo é condicionado para responder da mesma forma da próxima vez que o mesmo incentivo ocorrer [Skinner, 58]. Esta definição é muito similar à definição clássica de Pavlov de comportamentalismo³. A contribuição da teoria de Skinner é baseada na ideia de que um indivíduo pode emitir uma

² Instrução Programada – apresentação de novos conceitos de uma forma gradual e sequencial de passos controlados ao aluno, que trabalha por si, ao seu próprio ritmo e após cada passo testa a sua compreensão respondendo a questões. a resposta correcta é-lhe fornecida bem como informação adicional.

³ Comportamentalismo - Pavlov, médico russo, quando alimentava os seus cães de laboratório introduzia o ruído de uma campainha. Por meio de medições comprovou que o som da campainha provocava salivação nos cães, mesmo que o som não viesse acompanhado de comida. Depois de anos avaliando os resultados experimentais, estes dariam lugar à teoria conhecida como condicionamento clássico de Pavlov, que seria mais tarde a base do comportamentalismo, escola psicológica que pretende explicar e prever o comportamento, que trás alguma luz sobre muitas das aprendizagens inclusivamente em humanos.

resposta sem um incentivo externo. A IP foi o método pioneiro que condicionou o aparecimento dos IAC. Devido à sua arquitectura baseada em janelas condicionou as primeiras tentativas para fornecer algum tipo de adaptabilidade na instrução. Apesar de ter sido usada em vários domínios com algum sucesso, nomeadamente em situações de treino, a sua fraca capacidade de adaptação, limitada pelas ramificações estáticas das suas janelas contribuíram para o surgir de algumas dificuldades, amplamente identificadas na literatura, [Silva, 98].

- Um estilo de comunicação imperativo e monodireccional computador→ aluno, onde o aluno assume uma atitude passiva e submissa;
- Um caminho linear e fixo a seguir, independentemente das acções dos alunos;
- A total não observância de quaisquer aspectos sociais, não considerando diferenças existentes entre os alunos como pessoas que são.

Os primeiros sistemas IAC (década de cinquenta) eram pois programas lineares, caracterizando-se precisamente por fornecer o conhecimento de uma forma linear ou sequencial. A ordem pela qual eram exibidos os conteúdos nunca era mudada, devendo todos os utilizadores (alunos) seguir o mesmo percurso dentro do curso, ou seja, dois utilizadores, que forneçam as mesmas respostas executam exactamente o mesmo percurso, independentemente da forma como interagem com o sistema, do seu grau de conhecimento do domínio em estudo, das suas crenças individuais.

Estes sistemas lineares adoptavam a teoria comportamentalista de B.F. Skinner, como já foi referido. O professor é a figura central do processo ensino-aprendizagem e o aluno é uma figura passiva, o qual deve compreender os conteúdos transmitidos pelo professor para num segundo momento responder às questões relacionadas com os conteúdos abordados. Nesta teoria o principal objectivo era obter comportamentos desejados a partir de estímulos. Esses estímulos ou reforços, poderiam ser positivos ou negativos, visando reforçar ou evitar determinado comportamento. Os sistemas IAC adoptavam apenas estímulos positivos, pois quando o estudante errava não deveria ser aumentada a força de acção que levou ao erro por meio de um reforço negativo, [Skinner, 58].

Eram desenhados para apresentar ao aluno um problema, receber e armazenar a sua resposta, e “classificar” a performance global do aluno na prossecução da tarefa. Não se preocupavam com a forma como os alunos aprendiam, assumindo que se o sistema disponibilizasse

determinada informação, o aluno a iria absorver, de forma correcta, passando esta a integrar o seu conhecimento.

A grande limitação destes sistemas deve-se, por um lado, ao facto de não ser realizada uma análise aprofundada das respostas do aluno, e por outro lado, ao carácter restrito do diálogo com o mesmo. Por outras palavras, não era possível raciocinar acerca das respostas do aluno. Não existia também forma de raciocinar acerca do conhecimento pedagógico, limitando as formas de ensino a uma escolha predefinida, [Silva, 98].

Dado que o texto estava codificado nas próprias janelas, não era possível dispor de várias formas de expressar um mesmo conceito sem criar novas janelas para cada conjunto de escolhas. O currículo, encontrava-se representado de forma implícita, ou seja, as matérias e a sua organização eram definidas desde o início. Não se trata, pois, de um currículo aberto, capaz de permitir um ensino orientado para o aluno, constituindo sobretudo uma simples organização ou sequência de janelas a apresentar numa determinada ordem, condicionada pelas respostas do aluno.

Os softwares baseados neste modelo comportamentalista apresentam alguns pontos fracos:

- O aluno é controlado pelo software; não é uma figura activa no processo ensino – aprendizagem;
- O aluno é direccionado a tomar atitudes frente a estímulos apresentados;
- Não há uma preocupação com o processo de raciocínio e aprendizagem do aluno;
- São utilizados artifícios de reforço como notas e elogios;

Os IAC apenas apresentam os conteúdos, não motivam o aluno nem fomentam a sua aprendizagem. Neste contexto o aluno é quem precisa de se adaptar ao sistema e não o sistema ao aluno e à sua forma de aprender.

A partir da década de sessenta, começa-se a dar importância às respostas dos alunos, visando fornecer um feedback mais adequado. Esses programas IAC que utilizavam respostas dos alunos para controlar o material de estudo eram chamados de programas ramificados. Foram desenvolvidos sistemas que alteravam a apresentação de novos conteúdos, baseados nas respostas obtidas anteriormente. Não sendo suficiente apresentar a informação sempre da mesma forma para todos os alunos, quem desenhava o sistema tinha ainda assim de antecipar

todas as possíveis respostas. Os programadores tinham de saber com antecedência que tipos de respostas seriam possíveis e decidir que informação o sistema deveria apresentar, prevendo assim todas as ramificações possíveis, daí a designação de programas ramificados.

Estes sistemas foram os primeiros a “modelar” o aluno, de uma forma limitada, pois apenas modelavam o seu comportamento e não tentavam sequer modelar os diferentes estados de conhecimento, através dos quais o aluno evolui.

Estes programas (bem como os programas lineares) possuíam um número fixo de tópicos, apenas se distinguindo pela capacidade de tomar determinada decisão em função da resposta a um determinado exercício.

Este tipo de sistemas evoluiu através do uso de técnicas de identificação de padrões para analisar as respostas do aluno e, mais tarde, para os chamados sistemas adaptativos (geradores), capazes de gerar o material didático, em particular para problemas de aritmética e de vocabulário.

Nos anos setenta surgiram então os sistemas IAC adaptativos. Eram sistemas primários no aspecto da adaptação, não possuindo nenhuma relação com os métodos e técnicas de adaptação hoje conhecidos. Apresentavam ao estudante problemas com um grau de dificuldade adequado às necessidades específicas de cada aluno, tentando facilitar o seu processo de aprendizagem. Nestes sistemas, o modelo do aluno baseava-se em parâmetros representativos do comportamento detectado, ao invés de usar uma representação explícita dos seus conhecimentos. Esta evolução, no entanto, representa um passo na direcção a um ensino individualizado, mas ainda insuficiente, uma vez que as técnicas utilizadas apresentavam vários problemas importantes, tais como:

- Dificuldade na análise e compreensão das intenções e das dificuldades do aluno (sendo difícil retirar informação sobre o aluno através das suas interacções);
- Incapacidade de raciocinar sobre o domínio de ensino, o qual era ainda um domínio muito restrito;
- O conhecimento pedagógico – o que ensinar e em que situação – não é representado numa forma que permita raciocinar sobre ele;
- O conhecimento do domínio está combinado com o conhecimento pedagógico, dificultando a manutenção de ambos.

Independentemente do tipo de IAC, todos eles sofriam dos mesmos problemas, tais como a mistura existente entre o tópico em estudo e a sua forma de exibição. Ou seja, a sequência do currículo e as técnicas pedagógicas estavam embutidas nas actividades/métodos de ensino adoptados, o que dificulta a adaptabilidade e a alteração de técnicas pedagógicas a usar para um mesmo conteúdo.

Durante a década de sessenta e setenta a IA desenvolveu-se de uma forma rápida graças à confiança nos avanços obtidos na capacidade dos computadores. Nessa mesma época, a psicologia cognitiva começou a ser influenciada pelas novas teorias de aprendizagem. O Comportamentalismo começa a ceder lugar cada vez mais ao construtivismo de Piaget que ao invés de adoptar reforços programados, permitia que o conhecimento fosse construído pelo aluno por meio de descoberta livre, [Piaget, 89].

Podemos pois enquadrar os IAC em três situações tipo:

- Softwares de exercício e prática, em que o estudante adquire competências específicas e permite um estudo relativamente adaptado ao próprio ritmo;
- Softwares tutoriais, que disponibilizam caminhos alternativos, possuem níveis de complexidade, desenvolvem memorização e são úteis na revisão de um tópico;
- Softwares baseados em simulação, que oferecem maior interactividade, auxiliam o professor e os alunos, de certa forma aproxima-se da construção do próprio conhecimento.

Estas três situações tipo enquadram-se nas três categorias referidas anteriormente, e que são sintetizadas na Tabela 1.

Tabela 1 – Categorias e Evolução dos IAC

IAC	Lineares	Ramificados	"Adaptativos" ou "geradores"
	1950	1960	1967
	Limitam-se a apresentar os conteúdos de forma sequencial e pré determinada	Prevêem ramificações, o programa segue numa direcção consoante a resposta do aluno, no entanto esses caminhos tem de ser pensados e embutidos na altura do desenvolvimento	Utilizam já algum tipo de adaptação (muito primária)

1.6. IIAC – Instrução Inteligente Assistida por Computador

Em pesquisas de IA a preocupação centrava-se em obter melhores formas de representação do conhecimento. Surgem os chamados sistemas de Instrução Inteligente Assistida por Computador (IIAC) que apresentavam uma estrutura formada por componentes distintos para trabalhar com domínios educacionais, utilizando as técnicas de IA que se julgavam mais adequadas ao processo ensino-aprendizagem.

Em 1970, Carbonell definiu um novo tipo de IAC, designado na literatura anglo saxónica por Intelligent Computer Aided Instruction. O objectivo deste novo tipo de sistema era proporcionar um ensino mais efectivo e personalizado do que aquele que era alcançado com os sistemas IAC tradicionais, através da utilização de técnicas de IA [Carbonell, 70].

Carbonell, com o seu projecto Schollar, um tutor de geometria, utiliza técnicas de IA para tentar incorporar diálogos de tutoria em programas IAC. Em 1973, Sleeman, introduz a concepção básica do que viria a ser chamado de Sistema Tutor Inteligente – STI, em que o principal objectivo de tal sistema é comunicar o conhecimento de que dispõe de forma efectiva e eficaz. Em 1974, Self assumiu uma arquitectura tripartida para um STI – “o quê” (conhecimento do domínio); “quem” – (modelo do aluno); e o “como” - (a estratégia de ensino/tutoria). A estrutura básica de um STI foi pois, estabelecida há já trinta anos, [Sleeman, Hartley, 73], [Self, 74].

Por esta altura a noção de Processamento de Informação ⁴(PI), foi introduzida, o que combinou perfeitamente com o interesse da comunidade de IA no processamento de linguagem natural e na linguística. PI surgiu como um paradigma dominante durante a década de 70 e inícios dos anos 80. Este paradigma concebia o conhecimento humano como um conjunto de processos de “caixas negras”, em vez de apenas um conjunto de respostas a estímulos externos. A precisão dos modelos de PI, forneceu a promessa de ser possível representa-los por programas de computador.

Nos sistemas IIAC, agora designados por STI, a inteligência reside na capacidade que estes sistemas têm de efectuar acções que não foram previstas explicitamente pelo programador. Estas acções representam a capacidade dos STI de se aproximarem ao conceito de saber que

⁴ Processamento de Informação – na psicologia cognitiva é uma aproximação ao objectivo de perceber o pensamento humano. Vê a cognição como sendo essencialmente computacional, com a mente a ser o software e o cérebro o hardware.

ensinar, quando ensinar e como ensinar. Estes sistemas devem possuir uma representação da matéria de ensino (conhecimento do domínio), da pessoa que é ensinada (modelo do aluno) e da forma como o ensino se deve desenrolar (estratégias pedagógicas), [Self, 74].

Em 1982, Sleedman & Brown escreveram uma revisão histórica a respeito dos sistemas IAC e criaram o termo “ITS – Intelligent Tutoring Systems” para dar nome aos IAC e diferencia-los dos sistemas IAC tradicionais. A ideia central dos primeiros STI era que cada aluno deveria aprender fazendo, e a representação do conhecimento obtido pelo aluno seria de vital importância surgindo a noção de modelo de aluno, sendo esse um dos módulos interdependentes que separam as diferentes funcionalidades de um STI. Classificaram os STI existentes como sendo “*computer based problem-solving, monitors, coaches, laboratory instructors and consultants*”, [Sleeman, Brown, 82]. Nesse artigo, encontramos pela primeira vez o uso do termo “student model” para descrever uma representação abstracta do aluno no sistema. Sleedman e Brown classificaram os modelos do aluno como sendo:

- **Sobreposição** – um modelo do conhecimento do aluno como sendo um subconjunto do conhecimento de um perito;
- **Diferencial** – idêntico ao anterior, mas focando as diferenças entre o conhecimento do aluno e do perito;
- **Perturbação** – representando as concepções erradas do aluno como variantes da estrutura processual correcta do perito, muitas vezes designadas por “*mal-rules*”, regras erradas que descrevem os erros do aluno relativamente ao conhecimento do perito.

As primeiras tentativas de modelar o conhecimento do aluno foram baseadas num modelo “Buggy”, inicialmente proposto por Brown and Burton. Nesse modelo, “bugs” são erros do estudante em tarefas discretas, tais como uma subtracção incorrecta na operação de subtracção. No seu sistema Debuggy, Burton identificou 130 erros comuns na subtracção. O desafio consistia agora na análise do espaço do problema representado pelas respostas do aluno e determinar qual o erro ou conjunto de erros que melhor se enquadraria nesse espaço, [Brown, Burton, 78].

Uma vez que os aspectos subjacentes às limitações dos sistemas IAC tradicionais, são semelhantes aos aspectos que motivaram o desenvolvimento de outros tipos de sistemas baseados em conhecimento, as vantagens que advêm da utilização de técnicas de IA, com o

intuito de simplificar e tornar mais eficiente o desenvolvimento de programas de instrução, tornam-se evidentes.

Os primeiros esforços de aplicação de técnicas de IA em STI centraram-se na representação da matéria objecto de ensino, a qual se encontrava codificada implicitamente nos primeiros sistemas IAC. Têm sido aplicadas várias técnicas de representação do conhecimento, tais como redes semânticas, regras de produção, guiões (scripts), entre outras. No entanto, o progresso mais importante deu-se com a representação explícita e separada da base de conhecimento do domínio. Este desenvolvimento permitiu facilitar a modificação da base de conhecimento sem necessidade de reformular todo o sistema.

A partir de meados da década de setenta, a investigação na área dos STI começou a desenvolver esforços para a análise do estado de aprendizagem do aluno. Geralmente, um bom conhecimento sobre o domínio de ensino não é suficiente para garantir um bom ensino. Sem o conhecimento acerca do que o aluno sabe e do que não sabe, das suas motivações, o sistema de ensino não pode oferecer uma instrução adaptativa. Assim, as técnicas de IA começaram a ser utilizadas para avaliar o estado de conhecimento do aluno, [Carr, Goldstein, 77], [Brown et al, 75].

Este conhecimento acerca do aluno pode ser utilizado por um módulo tutor para tentar decidir acerca da próxima estratégia de ensino a usar. Finalmente, as técnicas de IA podem ser usadas para representar o conhecimento pedagógico. Nos sistemas IAC tradicionais, as estratégias de ensino encontram-se embutidas no código do programa. Os procedimentos que implementam as estratégias de ensino estão organizados através de estruturas de ramificação; o próximo ramo ou procedimento a usar depende da resposta do aluno. Se o sistema necessitar de conter todas as respostas possíveis do aluno, o seu desenvolvimento tornar-se-á demasiado complexo, ou mesmo impossível. Por outro lado, a tarefa de manutenção torna-se difícil porque o conhecimento do perito foi incluído no código do programa na fase de implementação.

Em Sledman and Brown, alguns assuntos de ensino relacionados com os problemas de desenvolvimento de STI são focados. Reconhecem que muita comunicação por parte de tutores humanos está implícita e expressam a esperança que os STI fornecerão um espaço para os teóricos da educação desenvolverem *“teorias de ensino/aprendizagem ainda mais precisas”*. Essa precisão é possível, sendo necessária para a implementação dessas teorias através de software. Discutem também a necessidade de construir ambientes que encorajem a aprendizagem cooperativa, reconhecendo no entanto que os investigadores (naquela altura)

sabiam muito pouco acerca de como essa cooperação se processa em ambientes naturais de aprendizagem, [Sleeman, Brown, 82].

Durante os anos oitenta, investigadores de IA continuaram a focar as suas pesquisas nos problemas de linguagem natural, modelos de estudante e na dedução. No entanto, esta área também atraiu investigadores que não pertenciam à área das ciências da computação, sendo o mais notável deles John Anderson. Anderson dedicava a sua investigação à ciência cognitiva, desenvolvendo a *ACT** – *Adaptive Control of Thought*, uma teoria de cognição, que viria a influenciar grandemente o desenvolvimento de STI. Na Tabela 2 são focados os princípios da teoria *ACT** e as suas implicações para os STI, [Anderson,83], [Corbett, Anderson, 92].

Tabela 2 – Princípios da Teoria *ACT** e implicações para os STI.

Princípios da teoria <i>ACT*</i>	Princípios correspondentes de tutoria
O comportamento de resolução de problemas é orientado para a prossecução de objectivos;	Comunicar a estrutura dos objectivos que esta por detrás das tarefas de resolução de problemas;
Os conhecimentos declarativo e procedimental são separados. As unidades de conhecimento procedimental são regras IF-THEN chamadas de produções;	Representa o conhecimento do aluno como um conjunto de produções;
A performance inicial de uma tarefa é conseguida através da aplicação de procedimentos gerais (fracos) a estruturas de conhecimento declarativas;	Fornecer instrução no contexto de resolução de problemas; Deixar que o conhecimento do estudante se desenvolva através de sucessivas aproximações à competência desejada;
As produções de tarefas específicas atingem-se através da aplicação de produções “mais fracas” ao conhecimento declarativo. Estas tarefas específicas estão na base de uma maior performance e eficiência;	Fornecer feedback imediato aos esforços desenvolvidos;
Como resultado da prática adicional, as produções podem ser encadeadas em produções de larga escala;	Ajustar os saltos de aprendizagem de acordo com os progressos de aprendizagem;
O estudante mantém o estado corrente do problema numa zona limitada da memória de trabalho;	Minimizar a carga da memória de trabalho;

Apesar de Anderson e os seus colegas investigadores terem criado a *ACT** como uma teoria cognitiva, acreditavam que esta era rigorosa o suficiente para que os seus princípios fossem testados através de software. Dois dos mais conhecidos exemplos são o Geometry Tutor [Anderson, 86] e o LISPITS - LISP Intelligent Tutoring System [Corbett, Anderson, 92].

LISPITS tenta modelar os passos necessários para escrever um programa em LISP. Compara depois os passos actuais que o aluno deu, com o seu próprio modelo. Corbett e Anderson chamam ao processo de monitorização e remediação “*knowledge tracing*”. O seu objectivo é um modelo “*master*” onde cada aluno domina 95% das regras para um dado conjunto de exercícios, antes de avançar para o próximo nível. Estes investigadores descobriram que os alunos que usavam o LISPITS completavam os exercícios do modelo “*master*” de uma forma consideravelmente mais rápida que os alunos que trabalhavam sozinhos, mas não tão rápido quanto os alunos que trabalhavam com tutores humanos.

O nome de Anderson tornou-se intimamente associado ao trabalho de STI, referindo alguns investigadores o termo “*Anderson – style tutors*”, talvez porque os seus sistemas foram dos poucos que foram usados em situações de sala de aula e não apenas como projecto de investigação.

Por volta dos anos oitenta, muito do entusiasmo da IA para criar computadores “que pensam” esvaneceu-se com o amadurecimento dessa área de investigação. Os investigadores dirigiram agora os seus esforços para tarefas mais prosaicas como a construção de sistemas periciais que funcionavam bem em domínios tais como a resolução e diagnóstico de problemas. Ao mesmo tempo, assim que os STI começaram a sair dos laboratórios de IA para as salas de aula e outros fins de instrução, começaram a atrair reacções críticas. Algumas das fraquezas dos STI começaram a evidenciar-se, assim que os investigadores se aperceberam que os problemas associados com o seu desenvolvimento eram bem mais difíceis do que inicialmente era previsto. Foram propostas várias arquitecturas para STI, como abordaremos mais tarde. Antes de analisar quais são os módulos que compõem um STI, convém analisar algumas definições:

Segundo Gamboa, “*Sistemas Tutores Inteligentes são programas de software que dão suporte às actividades de aprendizagem*”. Esta definição, embora recente e expondo o objectivo final de um STI, não enfatiza correctamente a diferença existente em relação aos sistemas IAC tradicionais. Um STI não fornece apenas suporte à aprendizagem, mas utiliza também técnicas que conferem um menor ou maior grau de inteligência ao sistema, o que não acontece com os IAC, [Gambôa, Fred, 01].

Em Villareal, os “*Sistema Tutores Inteligentes simulam um tutor autoritário que possui uma estratégia de ensino dos conceitos do domínio do tipo um para um. É um perito num determinado domínio de conhecimento e actua como guia, tutor ou treinador. Este tutor deve*

poder adaptar-se às necessidades que surgem ao longo da interacção numa sessão de tutoria com o aluno”, [Villareal, Giraffa, 01].

Já a definição de Wenger aborda outro aspecto ao colocar que os “*Sistemas Tutores Inteligentes são sistemas instrucionais baseados em computador com modelos de conteúdo instrucional que especificam 'o que' ensinar, e estratégias de ensino que especificam 'como' ensinar*”. Começa a ficar claro aqui a utilização de módulos separados, contendo um modelo do domínio a ser estudado e de um modelo instrucional ou pedagógico, [Wenger, 87].

Em [Guardia, 93], um “*sistema tutor inteligente é um sistema de ensino assistido por computador, que utiliza técnicas de Inteligência Artificial, principalmente para representar o conhecimento e definir uma estratégia de ensino; é capaz de comportar-se como um perito, tanto no domínio do conhecimento que ensina (mostrando ao aluno como aplicar o dito conhecimento), como no domínio pedagógico, onde é capaz de diagnosticar a situação em que se encontra o estudante e de acordo com esta, oferecer uma acção ou solução que lhe permita progredir na aprendizagem*”.

Entretanto, uma definição mais detalhada, que incorpora um vínculo com a área de IA, é apresentada em [Fowler, 91]:

"Os Sistemas Tutores Inteligentes são programas de computador com propósitos educacionais e que incorporam técnicas de Inteligência Artificial. Oferecem vantagens sobre os sistemas CAI, pois podem simular o processo do pensamento humano para auxiliar na resolução de problemas ou em tomadas de decisões."

Reva Freeman generaliza o conceito ao definir um STI como “*qualquer programa de computador que contém alguma inteligência e que pode ser usado em aprendizagem*” , [Freeman, 00] .

Poderemos pois considerar um STI como um programa dotado de inteligência, recorrendo a técnicas de IA, que consegue iniciar/desenvolver/interferir e avaliar o processo ensino-aprendizagem, definindo o “que ensinar”, “quando ensinar”, e “como ensinar”.

1.7. Características dos Sistemas Tutores Inteligentes

Conforme se pode observar pelas definições anteriores, a principal característica de um STI é sem dúvida a inteligência embutida no próprio sistema. Tal inteligência está presente na forma de:

- Estratégias de aprendizagem (modelo pedagógico), que determinam como transmitir o conhecimento ao aluno;
- Informações a respeito do nível/estado de conhecimento do aluno (modelo do aluno), que permitem personalizar a instrução;
- Conhecimento de especialistas no assunto (modelo do domínio), que determina a abrangência daquilo que deverá ser transmitido ao aluno;

Os STI devem ter a capacidade de aprendizagem, característica necessária para se poderem considerar inteligentes. Segundo Jonassen, um STI necessita de satisfazer três testes para ser considerado inteligente, [Jonassen, 88], [Jonassen, Wang, 93]:

- O conteúdo do curso ou domínio do conhecimento deve ser codificado por forma a que o sistema possa aceder às informações, fazer inferências ou resolver problemas;
- O sistema deve ser capaz de avaliar se o aluno adquiriu o conhecimento desejado;
- As estratégias de ensino (*objectos de aprendizagem*), estratégias de tutoria, devem ser projectadas para reduzir a distância entre o conhecimento do especialista e o conhecimento do aluno.

Os STI devem também cumprir um conjunto de características segundo [Guardia, 93]:

- Devem ser “inteligentes” em comparação com os sistemas tradicionais de instrução por computador (IAC), sendo a mais valia os métodos de IA que devem incorporar;

- Devem possuir a capacidade tanto para resolver o problema proposto a um aluno, como também a capacidade de explicar como o resolveu;
- Em relação aos sistemas IAC tradicionais, permitem uma maior individualização da instrução, chegando mais longe através do entendimento das metas e “crenças” do estudante;
- Usam-se técnicas de IA para o planeamento, a optimização e a pesquisa, deixando que o sistema decida a ordem de apresentação dos conteúdos ao aluno;
- A interacção pode ser muito variada num STI: desde sistemas passivos (que esperam que o aluno realize uma acção), até sistemas que constantemente apresentam novas informações (tutor oportunista), com casos intermédios através dos quais transmite um conceito num dado momento ou eventualmente só quando o aluno o solicita;
- Não basta indicar um erro ao aluno, o sistema deve colocar hipóteses baseadas no historial de erros do aluno e detectar a fonte de tais problemas.

Outra característica recente é a utilização de interfaces mais sofisticados, fazendo uso de tecnologias como por exemplo a multimédia e a *Web*, que permitem atender a um público maior devido ao alto grau de disponibilidade e acesso, e à facilidade de utilização embutida no próprio interface.

Mas actualmente, um STI é considerado realmente inteligente se é capaz de identificar necessidades, motivações, desejos e características do utilizador, com o propósito de efectuar um processo de ensino - aprendizagem personalizado e efectivo.

1.8. Conclusões

Ao longo deste capítulo foi abordada a evolução dos sistemas de ensino baseados em computador, até ao que hoje se designa como STI. Sugeriu-se a necessidade de investigação em diferentes domínios que não apenas a Informática, dada a natureza multidisciplinar dos STI, desde logo a Psicologia e as Ciências da Educação, dado que se pretende intervir no processo de ensino. As características das diferentes fases de evolução dos STI foram referidas e sintetizadas realçando o papel fundamental da IA para tornar possível a adaptabilidade de tais sistemas. OS STI surgem como potenciadores do processo de ensino e não como substitutos do tutor humano, ficando exposto que a utilização de STI produz resultados interessantes relativamente a grupos de alunos que não os utilizam, mas ainda assim inferiores aos resultados obtidos por grupos de alunos com tutoria individualizada.

2. Arquitecturas de Sistemas Tutores Inteligentes

Neste capítulo é abordada a estrutura típica de um STI e algumas arquitecturas propostas por investigadores são referenciadas, sendo de seguida abordados os três módulos principais da arquitectura dita clássica: Modelo Tutor, Modelo Domínio e Modelo Aluno, bem como o Modelo de Interface. Alguns sistemas desenvolvidos são brevemente referidos.

2.1. Estrutura Típica de um Sistema Tutor Inteligente

Nos Sistemas IAC todos os componentes necessários ao processo de instrução estavam combinados numa única estrutura, o que causava problemas quando era necessário proceder a alterações no sistema, a diferentes níveis. Surgiu a necessidade de separar o sistema em componentes que representem as diferentes formas de actuação do tutor e do aluno numa situação de ensino:

- O conhecimento a ser ensinado;
- O módulo de instrução;
- Um mecanismo para modelar o aluno;
- O método de comunicação.

Clancey e Soloway sugerem os seguintes modelos para uma arquitectura de um STI, [Clancey,Soloway, 90]:

- Um modelo explícito do domínio e um programa capaz de resolver problemas do domínio;
- Um modelo do aluno que caracterize, com um nível de detalhe adequado, o que o aluno sabe;
- Um modelo de ensino capaz de fornecer instrução para remediar erros e/ou apresentar novo material didáctico.

O processo de ensino - aprendizagem necessita de uma interacção entre os diferentes agentes envolvidos: o tutor, o aluno, o perito do domínio e o ambiente de ensino ou interface. Vários autores, tais como Sleeman e Brown e Burns e Capps, argumentam que um STI deve conter quatro módulos interligados, como apresentado na Figura 2, [Sleeman, Brown, 82], [Burns,Capps, 88]:

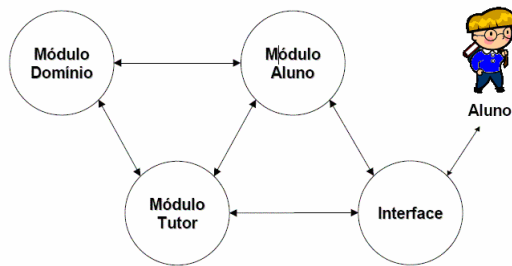


Figura 2 - Estrutura típica de um STI

Esta arquitectura é denominada de clássica, sendo também conhecida como funcional tripartida ou arquitectura tradicional de STI. O termo tripartida refere-se às funções associadas aos módulos tutor, do aluno e domínio. Esta proposta trouxe grandes avanços à modelação de ambientes educacionais, pois separou o domínio da sua forma de manipulação (no sentido de utilização). Permitindo, assim, que estratégias de ensino fossem associadas em função das informações oriundas da modelação do aluno. Ou seja, a arquitectura proposta deu início a uma nova forma de modelar os STI, onde o domínio estudado fica separado das estratégias, permitindo a troca das mesmas conforme a teoria cognitiva predominante no STI, ou em sistemas mais evoluídos, conforme a teoria cognitiva mais adaptada ao aluno.

2.2. Arquitecturas de Sistemas Tutores Inteligentes

É difícil identificar claramente uma arquitectura geral para os STI, no entanto vários trabalhos apontam para a arquitectura clássica referida anteriormente como a arquitectura tradicional de um STI. Avanços nas ciências da computação têm levado ao aparecimento de novas arquitecturas, existindo sempre no entanto, uma certa relação com a arquitectura dita clássica, mesmo na abordagem deste problema através do uso de agentes.

Assim, originalmente os STI implementavam-se seguindo o paradigma procedimental. Padeciam, no entanto, de algumas limitações relativas às linguagens utilizadas e ao hardware disponível. A impossibilidade de representar domínios complexos e a falta de visão global acerca das concepções das aprendizagens, não permitiam a criação de um sistema que pudesse ter as características básicas dos STI propostos por Carbonell, e que fossem eficientes. Surgiram também implementações usando programação orientada a objectos, as implementações através de módulos integravam-se para formar um sistema expansível. Por último, surgiram as evoluções baseadas em sistemas multiagentes, e que presentemente se revelam como capazes de melhor atingir os objectivos propostos, [Carbonell, 70], [Russell, Norvig, 03].

A modelação de STI é pois uma tarefa complexa, obrigando a considerar os três módulos fundamentais da arquitectura tripartida, proposta por Carbonell e revista por Self, de forma integrada e com tarefas concomitantes, [Carbonell, 70], [Self, 99].

Ao modelar um STI devemos considerar as características do domínio (conteúdo), o comportamento observável e mensurável do aluno, nas suas mais variadas vertentes (modelo do aluno) e, o conjunto de estratégias pedagógicas a serem utilizadas pelo módulo tutor na busca de um ensino personalizado e efectivo, potenciadoras do processo ensino – aprendizagem.

A sistematização do conhecimento pedagógico a ser colocado no tutor constitui a tarefa mais complexa de toda a arquitectura, no que concerne à modelação e implementação. No entanto, desde o início da pesquisa em Inteligência Artificial e sua aplicação à educação, a necessidade de inserir mecanismos de apoio à aprendizagem, tais como estratégias e táticas de ensino, baseadas naquelas utilizadas pelos professores nas salas de aula, é um dos grandes desafios na pesquisa desta área. As limitações de hardware e software são factores que contribuem para que esta tarefa não seja atingida ainda na sua plenitude, [Silva, 98].

Com o surgir de novas tecnologias, estas permitiram aos investigadores superar muitas questões em aberto na investigação de STI. A tecnologia de agentes tem-se mostrado muito promissora para modelação e implementação de STI, [Giraffa, Viccari, 99].

A arquitectura dos STI é normalmente composta de módulos responsáveis por representar com maior ou menor grau de detalhe as principais funcionalidades de um STI. Esses módulos são responsáveis pela representação do conhecimento a ser transmitido, pelas regras que determinam como esse conhecimento será transmitido, e pela representação do estado mental actual do aluno. Um exemplo é a arquitectura proposta na Figura 3, [Burns,Capps, 88]:

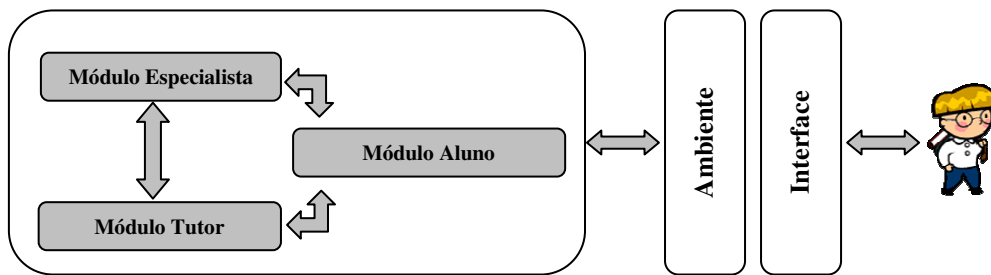


Figura 3 - Arquitectura de um STI, adaptado de [Burns,Capps, 88]

A arquitectura observada na Figura 3 possui três componentes:

- Módulo Aluno
- Módulo Tutor
- Módulo Especialista.

Essa arquitectura tripartida deu início a uma nova forma de modelar um STI, onde o domínio em estudo fica separado das estratégias de ensino, permitindo a troca dessas estratégias, conforme a teoria cognitiva predominante no STI, ou consoante as necessidades, motivações e adaptabilidade do aluno.

Módulo Aluno: responsável pelo diagnóstico do estado de conhecimento do aluno e que está armazenado no seu modelo. Esse modelo, consiste em informações persistentes sobre o estado mental actual do aluno, representando tanto o nível de conhecimento quanto informações referentes a preferências e objectivos individuais;

Módulo Tutor: selecciona as estratégias de ensino-aprendizagem que serão utilizadas para colmatar as deficiências de conhecimento do aluno, conforme as informações obtidas a partir do seu modelo;

Módulo Especialista: contém o conhecimento do domínio, armazenando as informações a respeito do assunto abordado pelo STI. Informações essas que são obtidas através de peritos e que serão utilizadas no processo de ensino - aprendizagem. O conjunto dessas informações compõe o modelo do domínio. Por outras palavras, contém o conhecimento a ser transmitido ao aluno.

A Figura 4 mostra como Mctaggart incorpora o interface como parte da arquitectura do sistema, totalizando quatro componentes, [Mctaggart, 01].

Self associou à arquitectura clássica um modelo de interacção, Figura 5, onde o módulo de domínio não é apenas uma forma de tornar as informações inter - relacionadas, mas sim um modelo dos aspectos do conhecimento sobre o domínio que o aluno pode aceder durante as interacções com o STI, Modelo da Situação, [Self, 99].

O módulo do aluno não relaciona somente as informações sobre a análise das interacções do aluno com o domínio, mas procura também uma maior contextualização dessas interacções em função das acções do aluno, do contexto em que elas ocorrem e da estrutura cognitiva do aluno naquele momento, Modelo de Interação.

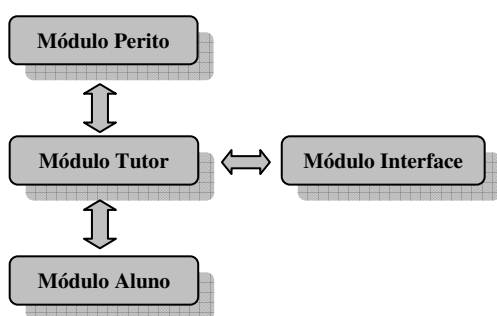


Figura 4 - Arquitectura de um STI proposta por McTaggart, adaptado de [McTaggart, 01]

O módulo tutor deixou de ser apenas o responsável pela selecção do conteúdo e das estratégias, para se tornar, de uma forma mais ampla, aquele que conduz o aluno de acordo com objectivos e desafios educacionais que o ambiente proporciona ao aluno, Modelo de Permissões.

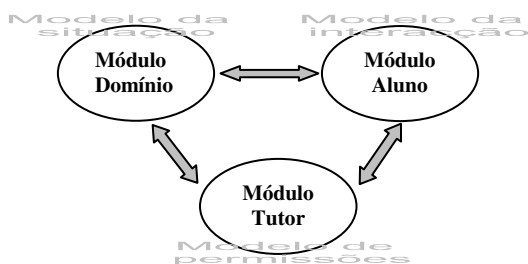


Figura 5 – Arquitectura de um STI segundo Self, adaptado de [Self, 99]

2.3. Módulo do Domínio

O Módulo do Domínio é uma base de conhecimento declarativo e procedimental relativo a um domínio específico. Alguns exemplos de formas de representação desse conhecimento são os sistemas baseados em regras e as redes semânticas. A tarefa de construir um modelo do

especialista que forneça a cobertura adequada do assunto que se pretende abordar é pois uma tarefa bastante difícil e dispendiosa.

Cabe resolver que o conhecimento armazenado nesse modelo é formado pelo conteúdo a ser apreendido pelo aluno, ou seja, por todo o material instrucional que será utilizado nas simulações, exemplos e explicações de um STI.

A evolução da multimídia educativa trata também da definição de modelos para especificação do domínio. Além da representação hierárquica normalmente adoptada, existe uma corrente de investigadores que adopta o conceito de mapas conceptuais para modelar o domínio.

Mapas conceptuais são instrumentos criados por Joseph Novak a partir das ideias do psicólogo americano David Ausubel. Os mapas conceptuais são representações de conhecimento por meio de uma rede de conhecimentos (nodos) interligados de acordo com uma determinada associação. Essas ligações podem ser multifacetadas, ou seja, pode haver mais de um tipo de ligação entre dois determinados nodos da rede, [Ausubel, 68], [Novak, Gowin, 84].

Um exemplo de um sistema que utiliza esta forma de modelação do domínio é o Metalinks. O sistema Metalinks tem *links* temáticos (não hierárquicos, associativos, ou “tangenciais”) além dos *links* hierárquicos pai-filho entre as páginas. Os *links* são categorizados para indicar o tipo de relacionamento que representam, sendo possível ao autor criar os seus próprios tipos de *links*, [Murray et al, 01].

2.4. Módulo Tutor

O Módulo Tutor (também chamado de módulo de instrução ou pedagógico) contém conhecimento para tomar decisões a respeito das estratégias de ensino que serão utilizadas no processo de ensino – aprendizagem. Basicamente, o Módulo Tutor é responsável por definir qual a informação que deve ser apresentada ao aluno, como e quando apresentá-la. Trata-se no fundo de definir as estratégias pedagógicas para potenciar o trabalho do aluno.

O conhecimento desse modelo afigura-se na forma de objectos de aprendizagem, que serão utilizados pelas estratégias e acções de ensino. A partir dessas estratégias, será definido a conteúdo a ser apresentado ao aluno e por meio das suas acções estabelece-se a forma de interacção. As estratégias visam estabelecer quando determinada acção será utilizada. A partir

das informações contidas no modelo do aluno, as regras das estratégias activam uma determinada acção que se pressupõe responder adequadamente ao estado mental actual do aluno.

A tomada de decisão a partir de uma estratégia visa definir qual deverá ser o próximo tópico a apresentar, e se houver mais de um tópico sugerido, qual a sequência considerada mais adequada.

Além de definir quando e o que apresentar ao aluno, o aspecto mais importante e difícil é determinar como o fazer. Daí a razão do módulo tutor também ser chamado de módulo pedagógico: a escolha da forma como os objectos de aprendizagem serão utilizados será guiada pela(s) teoria(s) pedagógica(s) embutida(s) no modelo. Idealmente, essas teorias deveriam ser acessíveis mas não embutidas, por forma a facilitar a escolha, sem qualquer tipo de condicionamentos.

Segundo Mizukami, as teorias pedagógicas podem ser classificadas utilizando características do processo ensino – aprendizagem, como sintetizado na Tabela 3, [Mizukami, Nicoletti, 86].

Os STI aplicaram as teorias descritas na Tabela 3, afim de obter um modelo tutor compatível com as suas características. De todas as teorias pedagógicas citadas, apenas duas se destacam como sendo amplamente utilizadas: a Teoria Comportamentalista e a Teoria Cognitivista. Os modelos criados a partir dessas teorias são normalmente denominados Modelo Comportamentalista e Modelo Construtivista, respectivamente, tendo sido já referidos anteriormente.

O Modelo Comportamentalista é composto por instruções programadas, propondo-se alcançar um determinado comportamento por meio do condicionamento, focalizando-se

no controlo externo e reforço. No Modelo Construtivista os alunos constroem as suas interpretações, interiorizando eventos externos.

Uma outra forma de incorporar características construtivistas é levar o aluno a criar estruturas de conhecimento enquanto percorre os tópicos que compõem a aprendizagem actual. Um exemplo disso mesmo é a utilização de mapas conceptuais como uma forma do aluno construir relações mentais entre dois conceitos.

Tabela 3 - Teorias Pedagógicas e o Processo Ensino-Aprendizagem, adaptado de [Mizukami, Nicoletti, 86]

Abordagem	Características do processo ensino-aprendizagem
<i>Tradicional</i> Snyders, G. (1974) Saviani, D. (1980)	Ênfase dado às situações de sala de aula, onde os alunos são "instruídos" e "ensinados" pelo professor; educação subordinada à instrução; conteúdos e informações têm de ser adquiridos e os modelos imitados; processo centrado no professor.
<i>Comportamentalista</i> Skinner, B. F. (1974)	Arranjo e planeamento de contingência de reforço aplicadas ao aluno; foco nas mudanças comportamentais úteis e adequadas; uso de condicionantes e reforços arbitrários para uma aprendizagem eficaz.
<i>Humanista</i> Rogers, C. (1972) Neill, A. L. (1963)	Professor como facilitador da aprendizagem; método não-diretivo: conjunto de técnicas que implementa a atitude básica de confiança e respeito pelo aluno.
<i>Cognitivista</i> Piaget, J. (1970) Ausubel, et al (1980)	Construtivismo baseado em interações do suíço Jean Piaget; aprendizagem implica assimilar o objecto em esquemas mentais; aprendizagem significativa: construção de novo conhecimento a partir de um conhecimento anterior; a aprendizagem só se realiza realmente quando o aluno elabora o seu conhecimento.
<i>Sócio-cultural</i> Freire, P. (1975)	Pedagogia construída com o aluno e não para ele; educação problematizadora ou consciencializada com o objectivo de desenvolver a consciência crítica e a liberdade.

As dificuldades de implementar um modelo totalmente construtivista estão relacionadas com as limitações de hardware e software, já que os sistemas capazes de "ver", "ouvir" e "sentir" o aluno ainda não estão comercialmente disponíveis. Com a difusão de tais sistemas, seria possível obter informações mais precisas sobre o perfil do aluno, o seu estado actual, as suas

reações e emoções com vista a determinar com maior exactidão o seu grau de motivação, por exemplo.

À medida que novas tecnologias permitam melhorar a recolha de informações sobre o utilizador, será possível um alto grau de precisão na sua modelação. Dessa forma, o Módulo Tutor poderá tomar as decisões mais acertadas, resultando numa maior eficiência do processo ensino – aprendizagem.

A ideia principal deste módulo é utilizar o computador como “ferramenta de aprendizagem” procurando automatizar o processo de ensino. Foi esta ideia que motivou os primeiros esforços na década de sessenta para fornecer suporte aos docentes. Desde então, os paradigmas utilizados na educação têm sofrido diversas mudanças, partindo de um estilo fortemente comportamentalista até aos dias de hoje onde o conceito de receber conhecimento é substituído pelo de descobrir e construir o conhecimento de forma significativa e activa.

Os STI reproduziam a forma de ensino que prevalecia no sistema educacional da época. Uma parte do conteúdo do domínio era apresentada numa ou mais janelas (apenas orientadas a caracteres, sem gráficos ou outros elementos didácticos) e a interacção limitava-se a pressionar a tecla “enter” para mudar de janela.

Segundo Wenger a visão clássica do papel do Módulo Tutor é tomar as decisões pedagógicas em função das interacções com o aluno. Estas decisões derivam de regras ou estruturas de conhecimento que representam o conhecimento do tutor relativamente ao domínio e estão representadas de forma explícita no sistema, [Wenger, 87].

Numa outra visão, o Módulo Tutor, é uma representação dos métodos que se usaram no tutor inteligente para fornecer informação ao estudante. Este modelo é complexo pois está pensado para dirigir o estudante no processo de ensino-aprendizagem e efectuar automaticamente ajustes nessa direcção à medida dos progressos do aluno, [Cho et al, 00], [Sierra et al, 03].

Num sentido prático, tem-se o seguinte problema para resolver quando se constrói o Módulo Tutor de um STI: O aluno manipula o Módulo do Domínio, e o Módulo do Aluno realiza inferências com base nessas manipulações. O tutor deve fazer uso dessa informação e fornecer ao aluno informação que seja realmente útil a este. Na sua forma mais geral, para poder definir correctamente a operação do Módulo Tutor, deve ser possível responder às seguintes questões:

- Quando é necessário instruir?

- Que tipo de instrução se deve fornecer?

São propostos os seguintes passos metodológicos para o desenho do Módulo Tutor, [Sierra et al, 03]:

- Analisar o modelo do aluno para definir claramente quais são as acções que este pode levar a cabo;
- Interpretar correctamente as acções definidas no passo anterior em função do tipo de conhecimento que o aluno deveria possuir para levar a cabo as ditas acções de forma correcta;
- Com base nos diferentes tipos de conhecimento identificados no passo anterior, determinar as estratégias de instrução mais apropriadas para que o aluno incorpore esses conhecimentos de forma significativa na sua estrutura cognitiva.

Na década de setenta Carbonell apresentou uma nova postura, que levou à prática através do sistema Schoolar, onde tomou em consideração a forma como o tutor estruturava o conteúdo do domínio na aula. O professor recebe o feedback dos alunos e tenta inferir o estado cognitivo actual do aluno, e a partir disso modifica o seu comportamento (através de estratégias de ensino) para que o aluno obtenha o maior benefício possível, [Carbonell, 70].

O Schoolar foi o primeiro programa para o ensino que utilizava uma representação do conhecimento baseada em IA. O seu principal objectivo era ensinar a geografia da América do Sul. Foi o primeiro sistema a reconhecer a possibilidade de separar o conhecimento da matéria a estudar, do resto da aplicação. Consta de um interface de linha de comandos tradicional, logo o Schoolar coloca perguntas sobre o tema e o aluno responde. Avalia-se a resposta e é gerada um retro alimentação. Dispõe também de um relógio para indicar ao aluno que esta a demorar demasiado tempo, [Carbonell, 70].

O sistema proposto por Carbonell utilizava redes semânticas ⁵para manter um diálogo com o aluno em formato de texto, utilizando um subconjunto de palavras da linguagem natural. Mais tarde apareceram trabalhos como WHY, o Sophie de Collie e Brown. Através da análise da aplicação Schoolar, Collins apresentou o projecto WHY, onde utilizou o método socrático de

⁵ Uma rede semântica é uma estrutura para a representação do conhecimento definida como um conjunto de nodos interligados por arcos rotulados. As redes deste tipo apenas captam definições dos conceitos, e proporcionam uniões entre conceitos.

ensino seguindo as directrizes propostas por Carbonell. Neste caso o tutor socrático ensinava através da exposição indirecta, mediante questões sucessivas necessárias para formular os principais princípios para poder analisar e avaliar as hipóteses, descobrir as contradições e finalmente realizar as inferências correctas, [Brown et al, 82].

Na década de oitenta, o crescente interesse pelos modelos didácticos computacionais diversificados ampliou o campo no conhecimento e entendimento do tutor humano, cuja complexidade não tinha ainda sido tratada computacionalmente. O tutor Quadratic podia modificar as suas próprias estratégias, ainda que sendo apenas uma modificação de parâmetros em lugar de uma troca efectiva de estratégias, [O'Shea, 82].

Mais tarde desenvolveu-se o sistema Predicate Logic Advisory Tool, denominado Plato, que ensinava matemática às crianças da primária. Plato gerava problemas e apresentava dicas ao aluno. Um dos seus módulos era constituído pelo jogo *"How the west was won"* que consistia em usar operações matemáticas para avançar; o jogo inclui West, um tutor inteligente que apoia o aluno na aprendizagem dos conceitos matemáticos associados ao jogo. Podemos ver este sistema como o primeiro sistema de aprendizagem completo. O paradigma utilizado é *"Issues and examples"* (temas pendentes e exemplos), onde se enumeram os tipos de combinações de operadores, a que se chamam "temas pendentes", ou seja, o que falta ao aluno aprender (esta é a informação que deve ser utilizada no modulo tutor). Espera-se então o momento oportuno para mostrar os exemplos de como aplicar o tema pendente e o benefício que isso representa. Este paradigma é aplicável, e podem construir-se STI de uma forma geral, estabelecendo os conceitos a ensinar (os "temas pendentes") e reconhecimento de padrões, para detectar se o estudante os domina ou não, [Castillo et al, 97].

Na década de 90 as contribuições das áreas como redes de computadores, sistemas distribuídos e comunicações proporcionaram novas ferramentas que se podiam aplicar aos STI, como por exemplo: email, foruns, de uma forma geral a Internet, com informação on-line, para além dos recursos didácticos importantes como vídeos, imagens, videoconferência, etc. Estas novas ferramentas permitiram revalorizar a aprendizagem assistida por computador. A Tabela 4 resume as principais características dos primeiros STI que surgiram desde a década de setenta até aos anos noventa, [Kinshuk, Patel, 96].

Tabela 4 - Características de STI desenvolvidos até à década de 90 – adaptado de Kinshuck, 1996

STI	Autor	Ano	Domínio	Características
Schoolar	Carbonell	1970	Geografia	Linguagem natural
Why	Stevens, Collins	1977	Meteorologia	Dialogo socrático
Sophie	Brown, Burton	1977	Electrónica	Linguagem natural
Wusor	Goldstein	1979	Estratégia	Jogo; Modelo de aluno: sobreposição
Guidon	Clancey	1981	Tutor Mycin	Primeiro sistema “glass-box”
West	Burton	1981	Estratégia	Treino e exemplos; modelo de aluno diferencial
Buggy	Brown	1981	Aritmética	Conhecimento incorrecto
Debuggy	Burton, VanLehn	1982	Aritmética	Diagnostico off-line
Steamer	Stevens , Hollan	1983	Desenho de caldeiras	Simulação, modelo mental
Lms	Sleeman	1984	Álgebra	Regras MAL
Meno	Wolf	1984	Programação	Gestão de discurso
Prosut	Jhonson	1984	Programação	Diagnostico de intenções
Actp	Anderson	1984	Tutor de Lisp	Modelação cognitiva
Sierra	VanLehn	1987	Aritmética	Previsão de erros
Sherlock	Lesgold, Katz	1991	Electrónica	Aprendizagem cognitiva
CIRCSIM	Evens, Michael, Rovick	1996	Medicina	Realmente utilizado no domínio médico

2.5. Interface

O Interface representa a interacção entre o STI e o aluno, devendo fornecer o melhor nível de interacção possível. É o ponto de contacto entre o aluno e o STI. É através dele que o STI apresenta os conteúdos a abordar e recebe as respostas do aluno, proporcionando a monitorização do seu progresso, pelo que se percebe facilmente que o interface condiciona de forma efectiva todo o sistema.

A quantidade e o tipo de características que são modeladas relativamente ao aluno dependem do tipo de interface utilizado. Esse é o principal obstáculo à perfeita representação do modelo

mental do aluno: saber que eventos disparados pelo aluno durante a utilização do STI são os mais adequados à definição do seu nível de conhecimento actual.

A proliferação do uso de interfaces baseadas na Web revela outros problemas inerentes a esse meio:

- A sobrecarga cognitiva deve ser controlada, visando evitar que o aluno fique desmotivado devido à grande profusão de elementos de percepção no interface;
- Outro problema advindo da interface Web é a necessidade de localização espacial, devido à não linearidade de navegação permitida. Para projectar um modelo de interface que evite estes problemas, proporcione uma experiência agradável ao utilizador e promova a aprendizagem, é necessário utilizar conceitos da área de Interação Homem-Máquina (IHM);

IHM é a área que estuda as interacções e relacionamentos entre humanos e computadores e caracteriza-se pelo seu aspecto multidisciplinar. O seu objectivo é desenvolver sistemas mais utilizáveis, úteis e amigáveis, fornecendo aos utilizadores sessões de interacção adequadas às suas experiências passadas, conhecimentos e objectivos, [Fisher, 01].

A aprendizagem do aluno num STI está directamente relacionada com a qualidade de uso encontrada no software. Os conceitos de IHM que medem a qualidade de uso de um software são: a usabilidade, a comunicabilidade e a aplicabilidade de um sistema.

A usabilidade está relacionada com a facilidade e a eficiência de aprendizagem e de uso, bem como com a satisfação do utilizador. A comunicabilidade procura avaliar o processo implícito de comunicação analista de sistemas -utilizador realizado por meio do interface. Já a aplicabilidade está relacionada com a flexibilidade de um sistema, em particular relativamente à sua utilidade diante de diversas situações diferentes, [Nielsen, 93], [Prates, De Souza, 00], [Fisher, 98].

2.6. Módulo Aluno

O modelo do aluno presente no Módulo Aluno, abrange tanto o conhecimento quanto o comportamento do aluno durante a sua interacção com o STI. Esse modelo é utilizado pelo Módulo Tutor para reconhecer erros, gerar e adaptar explicações e/ou conselhos, gerar

problemas, e acompanhar o progresso do aluno no currículo do curso (ao longo do curso). Deve representar o conhecimento do aluno e as suas competências cognitivas num determinado instante. Logo, é dinâmico, já que o conhecimento adquirido pelo aluno está em contínua transformação. Essas informações são necessárias para que o STI possa decidir qual “pensa” ser a melhor estratégia de aprendizagem que deverá ser utilizada, em cada momento. A cada interação do aluno com o interface do STI, o seu modelo é analisado e pode sofrer alterações, conforme as regras definidas no Módulo Tutor.

As diferentes formas como as informações do aluno podem ser confrontadas com o Modelo do Domínio (Modelo do Perito) geram uma classificação dos modelos de aluno, que se recorda a seguir, [Sleeman, Brown, 82]:

- **Modelo de Sobreposição:** um modelo do conhecimento do aluno como sendo um subconjunto do conhecimento do perito. A estrutura de representação deve ser a mesma nos dois modelos. Este modelo é o mais simples de todos, pois pressupõe que possíveis erros do aluno são devidos apenas à ausência de informação (*missing concepts*) no modelo do perito. No entanto, sabemos que comportamentos incorrectos podem ser decorrentes de concepções incorrectas (*misconceptions*) por parte do aluno;
- **Modelo Diferencial:** similar ao modelo de sobreposição, mas com o foco nas diferenças entre o conhecimento do aluno e do perito. Desse modo, o conhecimento do aluno é dividido em duas classes: o conhecimento que se espera que o aluno tenha e o conhecimento que não se espera que ele possua. Aqui também o conhecimento do aluno é um subconjunto do conhecimento do perito;
- **Modelo de Perturbação:** representação de concepções incorrectas (*misconceptions*) como variantes da estrutura procedimental das capacidades do perito, consideradas correctas. Essa relação de regras indevidas (*bugs*) descreve as concepções incorrectas do aluno em comparação ao conhecimento do perito.

Devido à sua simplicidade, diversos sistemas adoptam modelos clássicos como o Modelo de Sobreposição, definido por VanLehn como sendo uma forma de modelar o aluno a partir de comparações feitas em relação ao modelo do domínio representado por um modelo do perito. Nas Figuras 6, 7 e 8 podemos observar a representação dos vários modelos, [Vanlehn, 88].

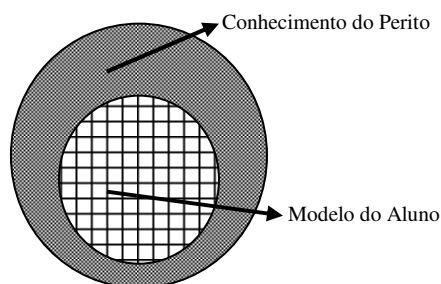


Figura 6 – Modelo de sobreposição

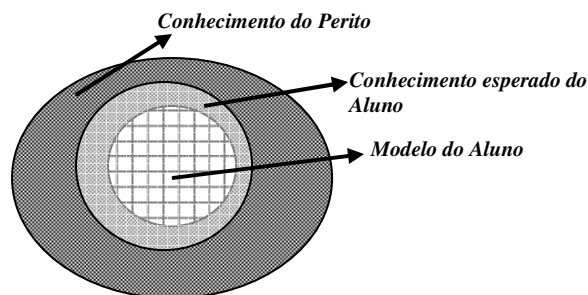


Figura 7 – Modelo Diferencial

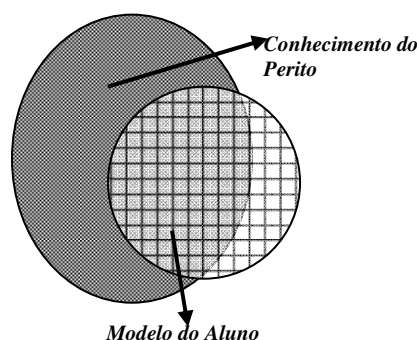


Figura 8 – Modelo de Perturbação

Deste modo, o objectivo é aproximar o modelo do aluno o máximo possível do modelo do perito, sabendo-se que ambos possuem a mesma estrutura.

Um STI obtém informações sobre o aluno através da sua interacção de forma explícita ou implícita. Quando o aluno responde a questões de exercícios ou problemas, o STI obtém uma declaração explícita do conhecimento actual do aluno sobre um determinado assunto. Outra forma explícita de obter essas informações é quando o aluno interage com o sistema, definindo as suas preferências, como por exemplo, ao ligar ou desligar a visualização de uma dica de auxílio. Além disso, a navegação realizada pelo aluno dentro do STI fornece implicitamente os interesses ou objectivos específicos dentro do domínio de conhecimento. Essas informações são utilizadas para criar uma representação do conhecimento actual do aluno, conforme um dos modelos de classificação vistos anteriormente.

Portanto, o modelo do aluno deve representar informações que sejam úteis no diagnóstico, visando a melhor selecção possível (óptima), de estratégias pedagógicas que apresentem a informação adequada ao nível de conhecimento do aluno, e da forma como este as prefere receber.

A necessidade de se desenvolverem sistemas personalizáveis capazes de adaptar a interacção com o utilizador conforme as suas necessidades deu origem na década de 70 às investigações sobre modelação do utilizador. Diversas áreas de aplicação serviram o propósito de experimentar e validar os métodos criados, destacando-se a área de STI.

Um modelo do utilizador, assim como um modelo do aluno, tem como finalidade representar as características de um utilizador específico, permitindo ao sistema adaptar a forma de interacção com o utilizador conforme as suas necessidades e preferências. Nos seus primórdios, a área de modelação do utilizador preocupava-se em modelar o maior conjunto possível de características sem levar em consideração o custo decorrente dessa abordagem. Actualmente, as pesquisas na área têm o seu foco naquelas características do utilizador consideradas indispensáveis.

As informações mais comuns em modelos de utilizador são, [Jameson et al, 99]:

- **Características Pessoais:** em sistemas onde características como idade, sexo e peso influenciam as respostas do sistema, tais como em aplicações da área médica, entre outras;
- **Interesses e Preferências:** num sistema onde o utilizador possa assinalar as suas preferências, por exemplo, quando o utilizador não possui o costume de solicitar ajuda e desactiva o assistente quando este surge, o sistema pode incluir essa preferência no seu modelo;
- **Competências:** quando o utilizador já conhece ou é especialista em parte do domínio da aplicação é importante que isso seja representado, permitindo que algumas regras sejam definidas para tratar esse caso. Por exemplo, num STI, podem existir estratégias pedagógicas que retirem do currículo do aluno a parte do domínio conhecida ou que exibam novos desafios ao utilizador com um grau de dificuldade superior, dado o seu conhecimento prévio (como veremos adiante, na secção de multimédia adaptativa...);
- **Objectivos:** a definição no modelo dos objectivos do utilizador pode ter um impacto considerável na interacção com o sistema. Por exemplo, num sistema de visita virtual de um museu, o utilizador aluno possui um objectivo diferente de utilizadores especialistas, ou de meros visitantes, permitindo que o sistema

varie a riqueza de detalhes ou a profundidade com que as informações são apresentadas;

- **Sistemas de Crenças:** a existência de conhecimento prévio e crenças do utilizador a respeito do domínio da aplicação pode influenciar a forma como o sistema conduzirá a interacção;
- **Padrões de Comportamento:** além das características cognitivas, existem pesquisas quanto à necessidade de representar os factores emocionais e motivacionais que influenciam a utilização do sistema pelo utilizador, visando aumentar o seu grau de satisfação com o sistema;
- **Contexto de Interacção:** as características referentes exclusivamente ao utilizador não são suficientes, sendo importante representar características do ambiente onde a interacção ocorre. Por exemplo, informações sobre a velocidade da ligação utilizada, sistema operativo, tipo de equipamento (*portátil, desktop*) são algumas características que permitem o sistema adaptar a apresentação e aumentar o grau de usabilidade do sistema;

Diversos métodos de modelação do utilizador são aplicados para representar características como as descritas acima. Dentre esses métodos, destacam-se os seguintes:

- **Aprendizagem Automática:** área da IA que estuda a criação de máquinas que aprendem, ou seja, de sistemas computacionais que consigam expandir o seu conhecimento a partir da análise de um grande conjunto de dados. Por exemplo, a observação do comportamento do utilizador gera um conjunto de exemplos de treino, visando prever as acções futuras do utilizador;
- **Modelos Estatísticos Preditivos:** a conjugação da Aprendizagem Automática com o raciocínio sob incerteza forma o que se denomina por Modelos Estatísticos Preditivos, tais como Árvores de Decisão, Redes Neurais e Redes Bayesianas. A utilização de componentes do modelo do utilizador que adoptam algoritmos estatísticos preditivos permite gerar previsões sobre o utilizador e adaptar a interacção com o sistema;
- **Modelos de Reconhecimento de Planos:** modelos capazes de inferir os objectivos do utilizador ou pelo menos parte dos seus planos para atingir os

seus objectivos. Desse modo, o sistema pode sugerir alternativas de acções, conforme os objectivos do utilizador;

- **Modelos Baseados em Lógica:** modelos compostos de conjuntos de crenças a respeito do utilizador a partir das quais podem ser feitas inferências. Regras podem disparar acções conforme o resultado das inferências, visando apresentar a informação adequada ao utilizador.

Os métodos acima apresentados prestam-se a diversas aplicações, entre as quais se destacam: Modelação do Aluno, Sistemas de Linguagem Natural, Multimédia Adaptativa, e Interação Homem-Máquina. A Modelação do Aluno, especificamente, ainda é uma área muito investigada. Alguns aspectos estão relacionados com a aprendizagem colaborativa, ao controle do aluno sobre a aprendizagem, e à possibilidade de inspecção e correcção do seu modelo, por exemplo, a partir de servidores de modelos de aluno. Esses sistemas agem como servidores disponibilizando funcionalidades para modelação do utilizador, [Fink, Kobsa, 00].

Uma outra questão pertinente é o estilo de aprendizagem, que pode definir-se como “*a forma pela qual um individuo aprende*”, e devido ao facto de que diferentes pessoas têm diferentes estilos de aprendizagem, estes reflectem as diferentes competências, interesses, debilidades e forças académicas. Diferentes investigadores apresentam formas distintas para reconhecer os estilos de aprendizagem dos alunos.

Uma definição de estilos de aprendizagem pode apresentar-se como “os rasgos cognitivos e fisiológicos que servem como indicadores relativamente estáveis, de como os alunos percebem interações e respondem aos seus ambientes de aprendizagem”. Requer-se portanto uma variedade de medições para avaliar os estilos de aprendizagem dos distintos alunos e determinar o seu potencial de êxito.

A maioria dos investigadores idealizou instrumentos para a detecção de estilos de aprendizagem dos alunos. Por exemplo, Felder definiu os seguintes grupos de classificação, [Felder, Soloman, 88]:

- **Tarefas activas – tarefas de reflexão:** forma de processar a informação;
- **Sensorial – intuitiva (racional):** forma de perceber a informação;
- **Visual – verbal:** forma de apresentar a informação;
- **Sequencial – global:** forma do processo de aprendizagem.

De um modo geral podemos afirmar que um STI deve pois abarcar o máximo possível de estilos de aprendizagem, por forma a maximizar a sua eficácia no processo ensino-aprendizagem, adaptando-se aos vários estilos dos alunos.

2.7. Conclusões

Algumas arquitecturas de STI foram abordadas, tendo sido explicitados os papéis de cada um dos módulos da arquitectura dita clássica, de forma a tentar fornecer uma visão global daquilo que se julga ser um STI e os seus componentes e funcionalidades. As características desejáveis num STI foram abordadas, tendo alguns STI desenvolvidos sido brevemente referenciados. Pode-se verificar que, os STI foram condicionados pelas correntes pedagógicas da altura em que foram desenvolvidos, procurando pois reproduzir o tipo de ensino em vigor à data. Várias definições para STI foram apresentadas, e no contexto deste trabalho sugerida uma delas.

Desenvolvimento de Sistemas Tutores Inteligentes

Neste capítulo irão ser abordados alguns Sistemas Tutores Inteligentes (STI) desenvolvidos nas últimas décadas, bem como as suas características e domínio de actuação. Algumas debilidades dos sistemas analisados referidas na literatura são focadas, assim como os pontos fortes de cada sistema. É utilizada uma tabela para sintetizar a informação sobre os STI e o capítulo finaliza-se com a referência a três STI, pela sua importância e actualidade, bem como pelo facto de actualmente serem utilizados.

2.8. Análise

As técnicas convencionais de programação para desenvolver STI, baseiam-se no desenvolvimento de módulos básicos, que interactivam entre si, e onde cada um possui a informação necessária para poder cumprir a sua tarefa. Esta definição de módulos dentro de um sistema tutor inteligente é útil na hora de identificar cada uma das suas partes, mas é de difícil implementação, que estes módulos estejam definidos completamente como indica a teoria que os suporta, já que existe sempre uma sobreposição com o domínio do problema a resolver, com todos os outros módulos.

A maioria dos STI não apresenta o nível desejado de inteligência. Independentemente das técnicas de programação utilizadas, a dificuldade de modelar como funciona a mente humana impede que muitos STI alcancem os seus objectivos. Vários investigadores asseguram que a modelação de STI através do paradigma de programação multiagentes é o melhor caminho para construir o modelo pedagógico, [Frigo et al, 04].

Tanto o paradigma dos agentes como o paradigma orientado a objectos partilham a passagem de mensagens para efectuar a comunicação e o uso de mecanismos de herança e de agregação para definir a sua estrutura, existindo no entanto divergências. Uma das principais diferenças tem a ver com o grau de autonomia. Os objectos têm controlo sobre o seu próprio estado de conhecimento, mas não têm controlo sobre o seu funcionamento. Os agentes controlam o seu

estado e o seu comportamento, são autónomos e podem recusar pedidos de assistência; os objectos não podem.

Uma outra distinção situa-se ao nível da comunicação. Nos objectos a definição dos mecanismos de comunicação faz-se através da indicação dos métodos que podem ser invocados, o que faz com que estes procedimentos sejam intrínsecos dos objectos que os desencadeiam. Nos agentes, o conceito de mensagem e da linguagem a utilizar é definido ao nível do colectivo ou comunidade de agentes, e não individualmente. Em termos de comportamento os agentes são flexíveis, cultivam a autonomia, percebem o seu ambiente e agem de acordo com os seus objectivos, [Novais, 03].

Uma das principais vantagens, segundo alguns investigadores, é a popularidade das metodologias orientadas a objectos, para as quais existem muitas metodologias para a implementação deste paradigma, como o caso de OMT (Object Modeling Technique), OOSE (Object Oriented Software Engineering), OOD (Object Oriented Design), RDD (Responsability Driving Design) e UML (Unified Modeling Language), entre outras. A experiência adquirida com a utilização destas metodologias desde pelo menos uma década pode dificultar a adopção das metodologias de agentes através da inércia gerada, um pouco como o modelo relacional restringiu a adopção de outros modelos de base de dados, [Iglesias et al, 01].

Portanto, independentemente do paradigma de programação que se utilize para a implementação de STI, procura-se respeitar a estrutura dos módulos, os interfaces e os distintos sub módulos que compõem a estrutura proposta por Carbonell ou qualquer um das suas versões posteriores, [Carbonell, 70].

Começando pelas primeiras implementações de sistemas que seguiam o modelo de Instrução Assistida por Computador (IAC), na década de setenta, aparecem implementações como o sistema SCHOOLAR que implementa a proposta de Carbonell, sendo esta a primeira implementação prática onde o sistema recebe um feedback dos alunos e tenta interferir no estado cognitivo dos mesmos.

Mais tarde, com o advento de técnicas da inteligência artificial (IA) e da ideia que o sistema não realiza a tarefa de ensinar de uma única forma, aparecem os tutores WHY e SOPHIE de Collins e Brown; também muito limitados na sua funcionalidade, mas foram estes que diferenciaram os IAC inteligentes dos STI, que até essa época se consideravam sinónimos, [Stevens, Collins, 77], [Villareal, Giraffa, 01].

O projecto WHY de Collins seguia as linhas propostas por Carbonell mas, explicando de modo socrático e por exposição indirecta. O projecto SOPHIE (Sophisticated Instructor for Eletronics) foi desenvolvido por Brown e Burton. O sistema possui três versões diferentes e possibilita a simulação de um circuito electrónico ajudando o aluno a identificar possíveis erros. O aluno recebe um circuito com um determinado erro e tem que determinar qual a peça defeituosa. O SOPHIE possui um gerador de hipóteses que simula as soluções sugeridas pelos alunos, testando se estão ou não correctas. Se o sistema considerar a hipótese do aluno inconsistente com os factos já apresentados, questiona o aluno e argumenta em contrário. SOPHIE tenta proporcionar aos alunos um caminho para a produção activa de estratégias e para a reflexão e auto – avaliação das mesmas. Utiliza um interface de linguagem natural. Todas as aproximações para resolver o problema na década de setenta eram muito rudimentares estando totalmente em desuso, [Stevens, Collins, 77].

A década de oitenta trouxe modelos mais diversificados. O tutor Quadratic, de O'Shea, podia modificar as suas estratégias pedagógicas, ainda que de modo muito limitado. Quadratic é um tutor sobre equações quadráticas baseado em regras de produção e que procura modificar continuamente essas regras de produção, proporcionando dessa forma algum tipo de adaptação. É baseado num modelo de instrução com quatro objectivos: aumentar o número de alunos com sucesso; melhorar a média de resultados; reduzir o tempo médio necessário para uma lição e reduzir o tempo de uso do computador. O sistema regista dados relevantes sobre o aluno e procura utilizar a estratégia mais adequada à situação, se não se verificar nenhuma melhoria em qualquer dos critérios referidos, o tutor pode rever a decisão e escolher outra estratégia. O'Shea enquadra o seu sistema tutor como utilizando a estratégia de aprendizagem por descoberta, mas por outro lado descreve a estratégia de tutoria como instrução planeada, referindo que a estratégia de ensino se baseia em fornecer aos alunos exemplos cuidadosamente escolhidos, que aumentam a probabilidade de o aluno descobrir uma determinada regra, [O'Shea, 82].

O Primeiro STI orientado para o ensino de linguagens de programação foi MENO, que estava especificamente orientado para o ensino da linguagem Pascal. Começou em finais dos anos setenta na Universidade de Massachussets em Amherst, e o seu principal objectivo consistia no diagnóstico de erros do tipo “não sintácticos” em programas simples de Pascal, para ligar esses erros com os conceitos errados que sobressaíam na mente do aluno. O sistema MENO utiliza uma série de “grupos” que representam as soluções possíveis para o problema apresentado. Possui ainda uma base de dados com os erros mais comuns que são cometidos pelos alunos, para que os possa reconhecer através duma análise do código fonte. De acordo

com os erros encontrados, apresentam-se ao aluno uma série de sugestões para corrigir o programa, indicando ainda quais são as causas dos seus erros e que conceitos aplicou de forma incorrecta, ou que conceitos desconhece, [Castillo et al, 97], [Wolf, 84].

Mais tarde apareceu o tutor chamado MENO-II, também na Universidade de Massachussets que estava orientado ao diagnóstico e análise das estruturas repetitivas, mas com a mesma base original que tinha o MENO tutor original. Nesta nova versão agregou-se um módulo denominado “Big Finder” que se dedica a analisar o programa do aluno dentro de uma árvore da solução proposta, e compara-o com uma árvore armazenada que descreve a solução de uma forma mais simples. Isto corresponde a uma melhoria relativamente ao uso de grupos, mas o esquema continua a ser demasiado rígido para poder encontrar erros para uma qualquer dada solução relativamente à solução possível, [Wolf, 84].

Outra das desvantagens dos tutores MENO, reside no facto de não existirem estudos empíricos para gerar o conhecimento de programação que o aluno possui, ou seja, não existia um módulo respectivo, de forma definida. Esta carência não deixa ver as diferenças entre os programadores menos experientes e os mais experientes, que podem adaptar as estruturas aplicadas noutras linguagens (como o C ou C++) para a resolução do problema original, [Jonhson, 86].

Neste contexto surge o tutor PROUST, analisando as debilidades dos MENO Tutor, reconsidera o problema da análise dos programas e agrega conhecimento da base de cada um dos alunos, ficando assim mais definido o módulo do aluno. Diferentemente do MENO, o PROUST está menos interessado em explicar as origens dos erros na programação do que na sua resolução, portanto possui uma estrutura mais similar a um conselheiro do que a um verdadeiro tutor. Logo, os métodos de diagnóstico de erros falham em reconhecer problemas do tipo não sintácticos que não são uma propriedade intrínseca do programa errado, mas são satisfatórios para detectar as intenções do aluno programador relativamente a estruturas e funções.

Definiu-se então, o “diagnóstico de intenções” como a grande motivação do PROUST. Isto significa que o tutor é capaz de deduzir as metas do utilizador e compara-las com o código que analisa, o que permite detectar erros não triviais na lógica da programação. Esta característica entendeu-se a outros domínios, fora do tutor PROUST, dada a sua grande utilidade, [Jonhson, 86].

Outro exemplo de um tutor inteligente orientado para o ensino de linguagens de programação é o COACH (Cognitive Adaptative Computer Help) de Selker. Neste, a ajuda e assistência ao aluno oferece-se na forma de um ambiente interactivo de ajuda, onde o estudante fornece um programa LISP como entrada e o tutor devolve informação relevante. A inovação que apresenta o sistema COACH, é a de um ambiente de ajuda interactivo e com um modelo do aluno adaptativo. Desta maneira o sistema vai evoluindo quando o aluno aprende e recebe distintos tipos de ajudas, dependendo do grau de conhecimento prévio do aluno. Selker, com o seu sistema COACH conseguiu criar ajuda personalizada a cada um dos alunos utilizadores, e por meio da observação das suas acções construir um modelo adaptativo do aluno. Para descrever os alunos, utilizou diferentes modelos, comparou os resultados entre os alunos “assistidos” com o COACH e os alunos não assistidos para medir o êxito do sistema como assistente de programação. O sistema COACH demonstrou ainda que o modelo do domínio era suficientemente adaptável para suportar domínios similares, como o caso do ensino de comandos do sistema operativo UNIX, [Castillo et al, 97], [Selker, 89]

Outro tutor de LISP é o ACTP, que é um dos precursores no uso da modelação cognitiva nos STI, não chegando no entanto a ser um tutor inteligente completo. Utiliza “grupos de comparação” que são bibliotecas de código comum, para encontrar os erros que os alunos cometem na hora de escrever código fonte em LISP. Este sistema podia detectar pequenos erros, mais comuns, em LISP, o que permitiu melhorar o conhecimento global dos alunos, [Russel, Norvig, 03].

O tutor *CAPRA* de Fernandez é outro tutor inteligente utilizado para o ensino de programação, foi desenvolvido no âmbito da investigação do diagnóstico e detecção de erros em ambientes de programação. Este tutor propõe lidar com as problemáticas anteriormente referidas mediante um tutor inteligente que possua as seguintes características, [Fernandez, 89]:

- Classificação heurística de erros;
- Analise de fluxo de dados;
- Execução simbólica.

Uma das características mais importantes que se pode observar no tutor *CAPRA* é que está construído em castelhano, pois foi desenvolvido em Espanha. Constitui uma das primeiras tentativas para desenvolver um tutor inteligente no idioma espanhol.

Outro exemplo de um tutor efectivamente utilizado é o GUIDON. Foi desenvolvido por William Clancey no Departamento de Informática da Universidade de Stanford. O Guidon pode ser considerado simplesmente como uma reorganização do MYCIN para fins didácticos. O MYCIN é um sistema de diagnóstico médico e prescrição, altamente especializado, destinado a ajudar os médicos nas infecções de meningite. O sistema é utilizado em faculdades de medicina para treino de alunos e médicos, possuindo além da base de conhecimento, as experiências de casos reais acumuladas pelo MYCIN, o que possibilita o fornecimento de bons exemplos para os utilizadores do sistema. A principal deficiência do sistema é pressupor que aluno possui o entendimento dos termos técnicos utilizados, bem como os resultados dos testes químicos apresentados.

O tutor GEOMETRY de Anderson, tem por domínio demonstrações de Geometria. O tutor compara as demonstrações do aluno com as suas e fornece as correspondentes pistas e conselhos, ou seja, cada aluno é avaliado pelo processo de resolução ideal dos métodos de prova matemáticos que estão armazenados no programa. a tecnologia utilizada por Anderson é baseada no seu modelo ACT*, já mencionado. É basicamente um modelo que compara os passos do aluno na resolução de problemas com a solução do perito. O tutor LISP da Carnegie Mellon University e o tutor GIL (Graphical Instruction in LISP), um tutor também de LISP mas com suporte gráfico, seguem também o mesmo princípio.

A grande maioria dos sistemas abordados são sistemas que não funcionam em modo gráfico, uma excepção é o sistema STEAMER de Hollan. O sistema tem por objectivo ensinar aos oficiais da Armada os problemas referentes à gestão de uma central de propulsão a vapor (comum a muitos barcos da marinha americana). O treino divide-se numa parte procedimental e numa outra parte conceptual sendo que a esta última, é dado um maior ênfase. A ideia principal é fornecer ao aluno uma concepção geral do que acontece numa central de vapor, possibilitando o exercício de previsão das consequências dos vários procedimentos, principalmente numa situação de potencial emergência. O STEAMER usa dois computadores, sendo que um mostra um diagrama da central de vapor e o outro apresenta informações sobre a central e o STI, que supervisiona a interacção do aluno com o sistema. Este não é considerado um sistema especialista directamente direccionado para o ensino. Na verdade representa um sistema de instrução capaz de fornecer uma simulação interactiva e observável do ambiente estudado.

Outro STI desenvolvido para a marinha norte americana foi o SHERLOCK e a sua evolução SHERLOCK II, cujo objectivo era treinar o pessoal de terra nos procedimentos de manutenção

de aviões a jacto. Um dado curioso é que as pessoas que usavam o SHERLOCK tinham uma performance superior que o grupo de controlo, e após 20 horas de instrução tinham um desempenho equivalente a alguns técnicos com experiência de trabalho.

Um exemplo de tutor inteligente desenvolvido em Portugal é o SPARSE-IT, (Sistema Pericial de Processamento Inteligente de Mensagens e Apoio à Análise de Avarias e à Reposição de Serviço), tem por objectivo fundamental auxiliar os operadores dos centros de controlo de energia eléctrica. Este apoio é especialmente necessário em regimes de exploração perturbada, em que o centro de condução poderá ser "inundado" por uma "avalanche" de alarmes que o operador terá que tentar interpretar rapidamente, de forma a minimizar os eventuais tempos de interrupção de serviço daí resultantes. A Figura 9 ilustra a arquitectura do sistema Sparse-IT.

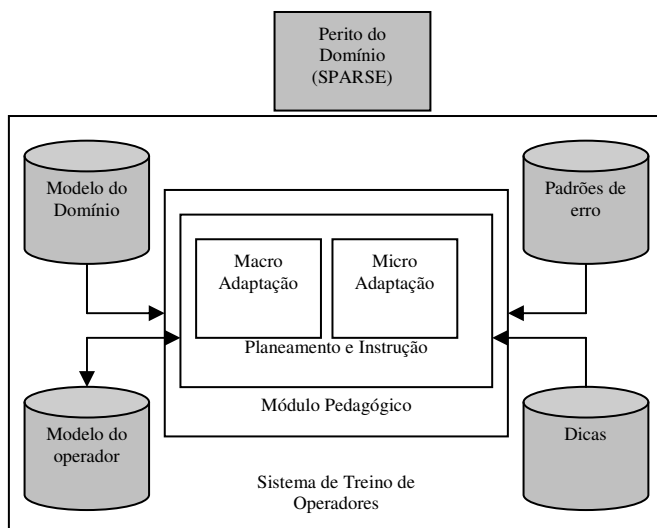


Figura 9 – Arquitectura genérica do tutor SPARSE-IT, adaptado de [Faria, 02]

Nos dias de hoje, deve destacar-se o projecto CIRCSIM, STI criado em conjunto pelo departamento de computação do Illinois Institute of Technology e pelo Departamento de Fisiologia do Rush College of Medicine. Este tutor está implementado e a ser utilizado no Rush College of Medicine para complementar as aulas teóricas sobre problemas cardiovasculares. Na sua base está a construção de um STI que tenta usar a linguagem natural para interagir com os alunos de medicina do primeiro ano. No CIRCSIM, os alunos resolvem pequenos problemas enquanto estão a ser tutelados de forma socrática pelo computador, [Cho et al, 00].

Dado ser um dos poucos STI que se utiliza efectivamente, será interessante referir as suas características e motivações para avaliar os factores de êxito e analisar a sua operacionalidade para “assistir” os alunos de medicina sem a supervisão directa de um tutor humano.

O CIRCSIM nasce como uma evolução do projecto HEARSIM, é codificado em Basic para Microsoft DOS. Diferentemente do HEARSIM, o CIRCSIM não armazena apenas as previsões correctas que realizava, como também a informação necessária para mostrar os resultados de cada um dos procedimentos. Do lado do aluno, este adquire factos e conceitos através das suas leituras, ao assistir às aulas e por participar noutras sessões de resolução de problemas. O CIRCSIM não explica os temas desde a sua base, o aluno já possui conhecimentos do tema quando utiliza o simulador. Desde então o programa é utilizado regularmente nos laboratórios de computadores, em sessões onde um tutor humano se encontra presente para interagir com os alunos em caso de problemas, [Rovick, Brenner, 83], [Rovick, Michael, 86].

Este tutor baseia-se num modelo qualitativo no qual intervêm sete parâmetros fisiológicos básicos. Os parâmetros e as suas relações causais formam um mapa de conceitos. Isto elimina o modelo do domínio, que ao ser fixo neste tutor, não requer modelação adicional. Os componentes básicos do tutor CIRCSIM são, [Evens et al, 01]:

- O “input understander”;
- O modelador do aluno;
- O planificador da sessão;
- O processador de discurso;
- O gerador de textos;
- A base de conhecimento/ solucionador de problemas.

Podemos de certa forma efectuar uma analogia com as arquitecturas referidas anteriormente em 2.2:

- Interface → O “input understander”, gerador de textos;
- Módulo do aluno → representado pelo modelador do aluno;
- Módulo tutor → o planificador da sessão, o processador de discurso e o solucionador de problemas;

- Módulo de conhecimento → a base de conhecimento;

Esta analogia é de certa forma incompleta, pois o CIRCSIM não está pensado como um STI multi-propósito, com todos os módulos completos, já que desde o início do seu desenvolvimento, sempre teve um domínio específico de actuação.

O seu funcionamento básico é o seguinte: apresenta-se um problema a um aluno, a resposta fornecida é comparada com a resposta do solucionador de problemas. Para continuar para o passo seguinte, o planificador de sessão usa as previsões do aluno e do seu modelo, para gerar um diálogo e detectar os possíveis erros do aluno. Este ciclo continua até se atingir o objectivo final do problema apresentado. O ciclo está implementado sobre uma máquina de estados finitos e os planos estão armazenados numa stack. Adicionalmente possui cerca de 750 lemas bem como um léxico básico, complementado com funções léxicas e gramáticas que se utilizam para gerar frases. Como é importante a correcção ortográfica das frases digitadas pelos alunos, para que o sistema as possa entender, existe um corrector que corrige automaticamente as palavras, que ou são mal escritas pelos alunos ou mal abreviadas. O gerador de texto produz orações de forma lógica através do planificador. As explicações podem fornecer-se de várias formas, por exemplo, de forma declarativa e de forma interrogativa, [Evens et al, 01].

No CIRCSIM, a resposta final a um problema proposto é única. Na maioria dos problemas de engenharia em geral, e nos de programação em particular, por exemplo, um problema pode admitir mais do que apenas uma solução satisfatória, existindo soluções prováveis e uma ou mais que se revelam como óptimas. Este dado complica sobremaneira a construção de STI para estes domínios.

O CIRCSIM possui excelentes resultados em experiências em aulas reais, carece no entanto de uma característica fundamental, que é escalabilidade. Só funciona correctamente no domínio de ensino para o qual foi projectado. Este problema é recorrente em muitos dos tutores analisados, já que a versatilidade foi sacrificada para melhorar o desempenho num determinado domínio em particular. Esta solução de compromisso pode ser uma solução válida no estado actual de desenvolvimento dos STI, no entanto a escalabilidade seria um marco teórico clássico nos STI, permitindo uma mudança de domínio mais ampla.

A Tabela 5 faz uma síntese de alguns tutores inteligentes, desenvolvidos nas últimas décadas.

Tabela 5 – Síntese de STI, adaptado de Park et al, 1987

Sistema	Domínio	Repr. do conhecimento	Modelo do Aluno	Estratégia do Tutor	Referência
SCHOOLAR	Geografia	Rede semântica	Sobreposição	Socrático	Carbonell, 1970
WHY	Tempo (Causas da chuva)	Guiões "Scripts"	Perturbação	Socrático	Stevens, 1982
INTEGRATE	Matemática (Integração simbólica)	Regras	Sobreposição	Ambiente reativo com orientação	Kimball, 1982
SOPHIE	Detector de falhas em circuitos electrónicos	Rede semântica	Sobreposição	Ambiente reativo com interações guiadas	Brown/Burton et al 1974; Burton, 1982
WEST	Expressões aritméticas	Regras	Sobreposição	Ambiente reativo com treino	Burton e Brown, 1979
BUGGY	Subtração	Rede procedimental	Perturbação	Ambiente reativo com orientação	Burton e Brown, 1979
WUSOR	Lógica	Redes semânticas	Sobreposição	Ambiente reativo com treino	Goldstein, 1982
EXCHECK	Teoria de conjuntos e lógica	Regras	Sobreposição	Ambiente reativo com orientação	Suppes, 1981
BIP	Programação em Basic	Regras	Sobreposição	Ambiente reativo com orientação	Barr et al, 1976
SPADE	Programação em Logo	Regras	Sobreposição	Ambiente reativo com treino	Miller, 1982
ALGEBRA	Matemática	Regras	Sobreposição	Ambiente reativo com treino	Lantz et al, 1983
LMS	Matemática	Regras	Perturbação	Ambiente reativo	Sleeman, 1982
GEOMETRY TUTOR	Geometria	Tem por base a teoria ACT	Sobreposição	Ambiente reativo	Anderson/Boyle et al 1985
QUADRATIC	Matemática	Regras	Sobreposição	Ambiente reativo com orientação	O'Shea, 1982
GUIDON	Medicina	Regras	Sobreposição	Ambiente reativo com interações estruturais	Clancey, 1982
PROUST	Programação em Pascal	Rede semântica	Perturbação	Ambiente reativo com orientação	Soloway et al, 1983

Sistema	Domínio	Repr. do conhecimento	Modelo do Aluno	Estratégia do Tutor	Referência
STEAMER	Propulsão a vapor	Modelo funcional	Sobreposição	Ambiente reativo com orientação	Williams/Hollan et al 1983; Hollan/Hutchins et al 1987
MACAVITY	Engenharia	Regras e "frames"	Sobreposição	Treino	Slater, 1985
MENO	Programação	Redes	Perturbação	Orientador	Soloway, 1983

Outros dois exemplos de sistemas também em funcionamento são o projecto ANDES, um tutor de Física e o AUTOTUTOR, que pode ser utilizado também para Física e no domínio da introdução à informática, [AutoTutor], [Vanlehn et al, 05].

O projecto ANDES teve início numa iniciativa conjunta da Universidade de Pittsburgh, e da Academia Naval dos Estados Unidos, sofrendo várias evoluções desde então. Alguns anos de experiência na Academia Naval dos Estados Unidos comprovam que o ANDES melhora significativamente a aprendizagem dos alunos. Ao contrário de muitos STI, onde o aluno apenas introduz a resposta a um problema, no ANDES os alunos completam vários passos, como sejam o desenho de vectores, de coordenadas, a definição de variáveis e a escrita de equações, sendo fornecido feed-back após cada passo. Andes também fornece pistas quando o aluno solicita ajuda, sobre o que está mal ou o que deve ser feito a seguir. A diferença relativamente a outros STI reside na granularidade de resolução dos problemas, no ANDES a granularidade consiste nos passos necessários para resolver um dado problema, noutros sistemas consiste na resposta ao dado problema.

O tutor ANDES implementa algumas características que se revelaram importantes no seu sucesso:

- Feed-back imediato;
- Ajuda aos erros dos alunos (What's Wrong Help");
- Ajuda e sugestões para o passo seguinte ("Next Step Help");
- Granularidade ao nível dos passos de resolução de um problema;
- Gera grafos de soluções.

O domínio de actuação do ANDES é a Física, no entanto pode ser aplicado a outros domínios, que envolvam a resolução de problemas através da produção de sistemas de equações e da sua resolução, como os domínios de engenharia e ciência, [Vanlehn et all, 05].

O AUTOTUTOR é um STI baseado na Web desenvolvido por uma equipa de investigadores interdisciplinar, envolvendo investigadores da área de Psicologia, Informática, Linguística, Física, Engenharia e Educação, do Instituto para Sistemas Inteligentes da Universidade de Memphis. Este grupo tem realizado investigação da área de tutoria, estratégias pedagógicas e suporte a linguagem natural. Estas investigações têm fornecido o conhecimento empírico e as bases teóricas para o desenvolvimento de STI que possam ajudar os alunos a aprender, através do diálogo em linguagem natural com o sistema, sobre um determinado assunto. Presentemente o AUTOTUTOR é utilizado de duas formas: uma versão destinada à aprendizagem de conceitos base de Informática e uma versão conceptual de Física. No entanto, pode ser adaptado para utilização em diferentes domínios, o que realmente constitui uma evolução relativamente a outros sistemas, na tentativa de separar completamente o domínio em estudo do próprio STI.

O AUTOTUTOR funciona através do estabelecimento de um diálogo com o aluno, aparece como um agente animado que funciona como um parceiro de diálogo do aluno. O agente animado utiliza voz sintetizada, expressões faciais e gestuais na sua interacção com o aluno, constituindo mais uma vez uma diferença fundamental relativamente a outros sistemas, pois procura simular totalmente as funções de um tutor humano. Os alunos são encorajados a responder com respostas articuladas e relativamente extensas, que exibam o seu raciocínio, em vez de fornecer respostas curtas e fragmentos de conhecimento. Para alguns tópicos existem gráficos e animações. O AUTOTUTOR está organizado em módulos:

- **“Animated Agent”**, um tutor virtual tridimensional, que comunica com o aluno através de voz sintetizada, expressões faciais e movimentos gestuais. Cada um destes parâmetros de comunicação pode ser ajustado para maximizar a eficiência do AUTOTUTOR como tutor e parceiro de diálogo;
- **“Authoring Tools”**, permitindo que peritos de várias áreas possam criar facilmente conteúdos que possam ser utilizados nas sessões de tutoria, sem a necessidade de possuir conhecimentos avançados em áreas de programação e técnicas;
- **“Curriculum Script”**, todos os problemas e respostas para um determinado domínio estão organizados num “Curriculum Script”, onde para cada problema

existe a resposta ideal, possíveis respostas correctas, concepções erradas que o aluno possa ter, pares pergunta-resposta que possam surgir, uma lista de conceitos importantes relacionados e todos os passos relacionados com o problema em estudo;

- **“Dialog Advancer Network”**, responsável por gerir o diálogo entre o AUTOTUTOR e o aluno, adaptando-se aos passos do aluno e respondendo em conformidade;
- **“Language Analyzers”**, baseados nos recentes avanços da linguística computacional, com o objectivo de melhorar e efectividade do diálogo e potenciar um diálogo de iniciativa mista, onde o tutor não se limita a responder ao aluno;
- **“Latent Semantic Analysis”**, um método estatístico para comparar a similaridade conceptual de dois blocos de texto. O AUTOTUTOR usa este módulo para comparar as contribuições do aluno com as respostas esperadas, armazenadas no “Curriculum Script”. Destina-se a “medir” a qualidade das respostas dos alunos, baseando-se em correspondências com respostas correctas e possíveis respostas erradas contidas no “Curriculum Script”;
- **“Question Answering”**, permite ao AUTOTUTOR responder a problemas relacionados com a questão em estudo mas que não têm respostas no “Curriculum Script”. Utiliza o “Speech Act Classifier”, um parser, e conceitos relevantes para classificar a questão do aluno numa categoria apropriada. O sistema extrai depois a resposta relevante de um repositório de texto;
- **“Speech Act Classifier”**, determina o propósito de um dado passo do aluno, para que o AUTOTUTOR possa responder apropriadamente e de forma flexível ao aluno. por exemplo, sem este módulo o sistema não seria capaz de saber que a frase “ “I do not know what gravity is” , é um pedido de informação e não apenas uma declaração do nível de conhecimento do aluno.

2.9. Conclusões

Neste capítulo foram referidos vários STI. Da análise efectuada ressalta que o caminho a percorrer ainda é longo, pois o nível de inteligência de tais sistemas ainda se encontra aquém do esperado. Verifica-se na grande maioria dos STI uma dependência do próprio sistema com o seu domínio de actuação, um dos factores que limita e muito a escalabilidade de tais sistemas. Exemplo disso é o sistema CIRCSIM, um sistema efectivamente utilizado e com níveis interessantes de sucesso. No entanto, encontra-se fortemente dependente do seu domínio de actuação, não sendo adaptável a outros domínios. Uma possível excepção é o AUTOTUTOR, com a sua aproximação por “Curriculum Scripts” e “Authoring Tools” que pretende especificamente a independência do sistema relativamente ao domínio em estudo. Este é o sistema que mais perto se encontra dos objectivos desejados para um STI.

Parte 2 – Nova Geração de Sistemas Tutores Inteligentes

Nesta segunda parte da dissertação são abordadas as características expectáveis de um Sistema Tutor Inteligente, referindo as funcionalidades que um tal sistema deve proporcionar. Serão depois referidos alguns dos problemas que persistem em alguns STI e também problemas que influenciam o seu desenvolvimento, contribuindo para que a disseminação de tais sistemas não seja ainda uma realidade tão efectiva como o desejável. Depois de contextualizadas as características, limitações e problemas de STI e do seu desenvolvimento, algumas Frameworks de desenvolvimento de STI são abordadas, como forma de introduzir e justificar a necessidade da Framework Desenvolvimento de Sistemas Tutores Inteligentes, (FD-STI), que é proposta neste trabalho. Depois de contextualizada a necessidade da Framework e brevemente explicada, esta é analisada e os seus componentes detalhados, referindo as contribuições que podem proporcionar para o desenvolvimento de STI.

FD-STI – Framework de Desenvolvimento de Sistemas Tutores Inteligentes

O desenvolvimento de Sistemas Tutores Inteligentes (STI) é uma tarefa complexa, como referido, pois o seu objectivo é participar no processo de ensino, o que implica ter pela frente muitos e diferentes alunos. As suas necessidades individuais diferem, e muitas vezes os próprios alunos não se apercebem sequer que necessitam de ajuda. Consequentemente, um tal sistema deve ser capaz de se adaptar e monitorizar dinamicamente cada aluno. A mera apresentação da informação não pode ser qualificada como instrução ou ensino. O desenvolvimento de sistemas de ensino envolve várias disciplinas como a psicologia, as tecnologias de interface homem-máquina, a representação de conhecimento, as bases de dados, a análise e desenvolvimento de sistemas, [Kinshuk, 02], [Negoita, Reusch, 04].

Espera-se que um STI eficaz realize eficientemente as seguintes tarefas:

- Apresentar ao aluno um conjunto de conteúdos ou competências que este pretende aprender, de uma forma que se adapte às suas características pessoais,

ao seu estilo de aprendizagem e características psicológicas, fornecendo o conteúdo mais adequado, da forma que se julga mais correcta, no momento adequado;

- Aconselhar o aluno na forma de adquirir as competências desejadas e ajuda-lo na preparação de um plano de estudo adequado;
- Cooperar com o aluno na monitorização do seu plano de estudo;
- Monitorizar os planos de estudo dos diferentes alunos integrados em processos de aprendizagem colaborativa, principalmente porque os alunos devem estar ao corrente de outras actividades e a colaboração com outros elementos (alunos, monitores) deve ser regulada;
- Analisar de forma inteligente e interactiva o que os alunos estão a fazer, fornecendo diagnóstico e ajuda em tempo real;
- Efectuar um plano de conteúdos inteligente, aconselhando o aluno na forma de melhorar a sua aprendizagem.

A necessidade de inclusão de aspectos pedagógicos no desenvolvimento de STI, centra-se sobretudo na necessidade de adquirir conhecimento acerca de múltiplas estratégias pedagógicas, separadamente da aquisição de experiência no domínio. Um STI moderno deve incorporar várias estratégias de ensino. Cada estratégia tem as suas vantagens/desvantagens específicas, relativamente a outras estratégias, sendo desejável o uso da estratégia que se julga mais favorável e que reforce o processo de ensino-aprendizagem para um dado aluno. A selecção da estratégia tida como a mais adequada depende de vários factores: o conhecimento do estado de conhecimento actual do aluno, o domínio em estudo, a motivação do aluno e as suas características afectivas. Alguns investigadores propõem actores pedagógicos, como agentes que cooperam com o aluno, de forma a facilitar a aquisição e modelação do conhecimento, um problema que é ainda um passo crucial no desenho de STI, [Frasson, 97].

Vários problemas são associados com as metodologias de desenvolvimento de STI, [Self, 99], [Bra, Brusilovsky, 99]:

- Cada aplicação é desenvolvida independentemente;
- A “experiência” do tutor é embutida no código dos sistemas individuais;

- Existe pouca reutilização de componentes como o modelo do aluno, o modelo de tutoria, o interface com o utilizador;
- Não existe uma linguagem normalizada para representação do conhecimento, nem um conjunto de ferramentas para manipular esse conhecimento.

Como forma de ultrapassar muitos dos problemas que vão surgindo no desenvolvimento de STI, algumas *Frameworks* têm vindo a ser propostas por alguns investigadores, apresentando-se de seguida algumas delas. Kinshuk e Patel propõem uma *Framework* Conceptual para Sistemas Tutores Inteligentes baseados na Internet, [Patel et al, 01], [Kinshuk, Patel, 97].

A Internet é apontada como catalisador para o desenvolvimento de sistemas de ensino, proporcionando uma vasta fonte de informação e oportunidades para desenvolver STI com diferentes estratégias pedagógicas. A grande maioria desses sistemas baseia-se no paradigma de Hipertexto (entendendo-se aqui Hipertexto num sentido mais lato – Hipermédia). Existiram algumas tentativas para desenvolver STI na Internet, diferindo estes sistemas dos tradicionais STI pelo facto de também se basearem no paradigma de Hipertexto em vez de utilizarem o paradigma dos Sistemas Periciais. O paradigma dos Sistemas Periciais tem por objectivo capturar a experiência humana substituindo o perito humano, fornecendo ajuda na análise de problemas e na sua avaliação. O paradigma de Hipertexto tem por objectivo proporcionar ao utilizador uma vasta gama de variedade de informação de diferentes fontes. Na sua forma pura, a implementação de STI recorrendo a Sistemas Periciais e a Hipertexto é respectivamente, fraca a nível de instrução e fraca a nível do paradigma construtivista, sendo estas falhas devidas a um hiato de contexto entre a metodologia utilizada e os propósitos do STI, pois as metodologias de engenharia de software não foram desenvolvidas especificamente para desenvolver aplicações de tutoria, não possuindo os atributos desejáveis para facilitar a tutoria inteligente. Um STI herda funcionalidades poderosas nos pontos de convergência entre os seus objectivos e as capacidades da metodologia utilizada, mas herda também um hiato de contexto nos pontos de divergência entre os propósitos das tarefas que se realizam num STI e os propósitos da metodologia.

Os sistemas Periciais possuem duas características que dificultam a sua aplicação em STI: a decomposição do conhecimento, a sua representação e as inferências realizadas são exclusivamente hierárquicas; todas as relações são estabelecidas através de raciocínio, faltando *links* explícitos de informação. Adicionalmente existe sempre um limite no nível de “perícia” que um tal sistema proporciona, apresentando a solução em vez de guiar o aluno através de

diferentes passos até à solução final. Um sistema Pericial é moroso a desenvolver, limitado na sua experiência e inflexível comparado com o desenvolvimento utilizando o paradigma de Hipertexto que proporciona um desenvolvimento mais rápido, alterações e actualizações ao sistema mais fáceis de realizar, em conjunto com uma mais fácil interligação de informação hierárquica e não hierárquica. No entanto, a flexibilidade e liberdade proporcionada pelos sistemas baseados em Hipertexto pode frequentemente resultar no facto de o utilizador se perder na cadeia de informação e de *links*, divergindo do seu objectivo inicial. O sucesso de um sistema destes depende grandemente da motivação e disciplina do utilizador. Para os autores, a conjugação destes dois paradigmas, com a Internet, abre perspectivas interessantes para o desenvolvimento de STI, reconhecidos que são os cinco atributos da Internet que a beneficiam o processo de ensino relativamente aos processos tradicionais:

- Comunicação muitos - para - muitos;
- Independência do local onde se encontra o aluno;
- Independência de tempo e hora;
- Comunicação multimédia;
- Interação mediada por computador.

Tendo em mente estes dois paradigmas e a importância da Internet, é proposta pelos autores uma *Framework* com o objectivo de desenvolver STI, reduzindo custos, e que sejam implementados e utilizados por um grande número de alunos, justificando a pesquisa e investimento nesta área. A *Framework* baseia-se em cinco noções:

- O Contexto do STI – a base da *Framework*, estando esta noção relacionada com os vários contextos de um STI, particularmente o papel central do professor como colaborador no desenho e implementação, as capacidades e limitações da tecnologia educacional utilizada, a natureza e constituição do conhecimento;
- O Paradigma de Sistemas Periciais – a tutoria envolve mais que a mera apresentação de informação e requer validação das acções do aluno, com feed back imediato para prevenir erros, e feed back assíncrono, para verificação periódica do estado de conhecimento do aluno;

- O Paradigma de Hipertexto – a flexibilidade proporcionada pelos mecanismos de links permite ao professor produzir *Appllets* de tutoria inteligente, efectuar ligações entre elas e construir um sistema maior e mais complexo;
- Noção e orientação a objectos – o conhecimento é visto como sendo uma rede de objectos de conhecimento, sendo as *Appllets* de tutoria inteligente desenhadas com base nesse conhecimento, efectuando uma ligação entre a estrutura do conhecimento e o paradigma de objectos empregue por linguagens como o Java;
- Interface Homem - Maquina – o aluno não deve ser obrigado a seguir caminhos pré definidos e executar tarefas supérfluas. O sistema deve ser intuitivo e simples de utilizar.

A consideração destas cinco noções, leva a uma implementação utilizando hipertexto com capacidades de *links* de informação hierárquicos e não hierárquicos, usando também o mecanismo de inferência dos Sistemas Periciais. A base de conhecimento está distribuída por um conjunto de *Appllets* de tutoria inteligente, onde a estrutura é também hierárquica e a ligação entre elas fornece uma estrutura não hierárquica. Esta estrutura proposta é bastante flexível pois o STI pode ser copiado e modificado para criar um outro, o mesmo acontecendo com as *Appllets*, sendo portanto o processo de desenvolvimento facilitado e mais rápido. O mecanismo de inferência utiliza o paradigma de Sistemas Periciais, sendo dividido em componentes mais pequenos, (os mecanismos de inferência de cada uma das *Appllets*), proporcionando eficiência e uma maior flexibilidade.

Uma outra *Framework* é proposta por El-Sheikh e Sticklen, para desenvolver STI incorporando reutilização, [El-Sheikh et al, 98].

Neste trabalho a reutilização assume um papel preponderante, pois segundo os autores, apesar da extensa investigação em STI, poucos foram efectivamente usados comercialmente. As razões apontadas prendem-se com o facto de os STI serem difíceis de desenvolver, sendo necessárias grandes quantidades de tempo e recursos financeiros. Existindo a necessidade de formas mais fáceis e acessíveis de desenvolver STI, apontam a reutilização como forma de o conseguir. O objectivo é desenvolver STI para um vasto domínio de aplicações e não apenas dedicados a um domínio específico, baseando-se na metodologia de tarefa genérica dos sistemas periciais. Pretende-se pois desenvolver uma arquitectura de STI que possa interagir com qualquer sistema baseado em tarefas genéricas, sendo um tutor efectivo no domínio de aplicação

do sistema pericial. Alguns problemas são apontados no desenvolvimento de STI, como o facto de cada aplicação ser desenvolvida independentemente das outras, a experiência do tutor estar embutida nas aplicações individuais, existir muito pouca reutilização de componentes do STI, como o modelo de aluno, o interface e o modelo tutor, entre diferentes aplicações, principalmente devido à falta de uma linguagem normalizada para representar o conhecimento, um interface standard para permitir às aplicações acederem a esse conhecimento e um conjunto de ferramentas que permitam aos investigadores manipular o conhecimento.

O conceito de tarefa genérica pode ser entendido como uma aproximação para desenvolver software reutilizável, em particular *shells* reutilizáveis para análise e desenvolvimento de sistemas baseados em conhecimento. Cada tarefa genérica é definida como uma combinação de:

- Uma descrição de entradas e saídas;
- Uma descrição da estrutura do conhecimento a ser seguida pela tarefa genérica;
- Uma descrição da estratégia de inferência a utilizar pela tarefa genérica.

Para desenvolver um sistema, seguindo esta aproximação é necessário efectuar a decomposição de um problema até que uma sub-tarefa do mesmo corresponda a uma tarefa genérica ou até que seja identificado um método para realizar a sub-tarefa. As instâncias de tarefas genéricas que foram identificadas são depois implementadas, construindo-se “blocos” a partir de *shells* de tarefas genéricas provenientes de uma biblioteca e obtendo o conhecimento do domínio apropriado para a estrutura de conhecimento da tarefa genérica que irá ser utilizada. Seguindo esta aproximação, é proposta uma *Framework* que possa interagir com um sistema baseado em tarefas genéricas, para produzir um STI com flexibilidade de adaptação a qualquer domínio, como se pode observar na Figura 10. O utilizador interage com o STI (para receber instrução, feedback) e com o sistema pericial (para resolver problemas e ver exemplos). Com esta aproximação, em vez de se implementar o módulo perito para cada domínio, o STI comunica com o sistema de tarefas genéricas para extrair o conhecimento necessário para o domínio em estudo. Isto facilita a reutilização do tutor, do interface e de componentes do modelo de aluno para diferentes domínios. Como desvantagem, verifica-se que a qualidade do conhecimento do tutor é afectada pela representação de conhecimento utilizada pelo sistema pericial.

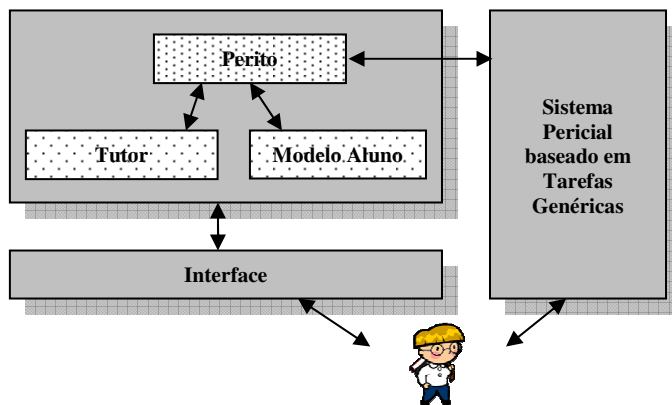


Figura 10 – Framework de Desenvolvimento de STI com reutilização – adaptado de [El-Sheikh et al, 98].

Lelouche e Ly propõem também uma *Framework* para desenvolvimento de STI, denominada ITS-FW, com o objectivo de facilitar o processo de desenvolvimento e torna-lo mais eficiente, através da reutilização, da flexibilização e da escalabilidade, [Lelouche et al, 03].

Nessa *Framework*, salienta-se a necessidade de envolvimento de diferentes áreas de pesquisa, e a necessidade de maximizar a reutilização, tentando reutilizar conceitos existentes noutras tecnologias de diferentes domínios, como por exemplo, desenvolvimento orientado a objectos e aplicações distribuídas, dado que as suas potencialidades e eficiência têm sido comprovadas em sistemas largamente utilizados. A *Framework* é apresentada como um conjunto de directivas e ferramentas para facilitar todas as etapas de desenvolvimento.

Como directivas propõem:

- A definição de uma representação normalizada para o conhecimento – sendo um esquema flexível e extensível capaz de representar e definir qualquer domínio de conhecimento. Baseando-se na notação XML, suportando diferentes tipos de estruturas de conhecimento tanto ao nível de meta-conhecimento como ao nível dos diferentes níveis de conhecimento, podendo ser representados sob diferentes formas: procedimentos, conceitos, regras de produção, proposições etc. a representação normalizada do conhecimento fornece um meio para definir e reutilizar o conhecimento ou meta-conhecimento, facilitando a partilha. O conhecimento sob um dado domínio pode ser transferido de um STI para outro.
- A definição de uma estrutura para localizar e aceder um determinado conhecimento do domínio, a partir de várias localizações. Pretendendo-se facilitar a distribuição do conhecimento de um dado domínio num ambiente

distribuído como a World Wide Web por exemplo. Desta forma, o conhecimento de um dado domínio pode ser acedido e reutilizado num outro domínio.

- A definição de uma forma de comunicação entre componentes, “Common Component Interface” – sendo esta capacidade comum e empregue em muitas outras tecnologias, definindo-se um conjunto de interfaces que explicitam a forma como cada módulo comunica com os restantes. Assim, qualquer componente pode comunicar com os outros, dentro de uma aplicação, desde que respeite o conjunto de interfaces pré – definido, facilitando por exemplo a integração ou substituição de um componente num dado STI.

Como ferramentas propõem:

- Um conjunto de operações de baixo nível, designado como conhecimento de runtime, ou seja, conhecimento procedimental contendo as operações mais simples como a adição, a subtração ou a comparação de dois valores, a representação de pontos e de segmentos de recta, que são normalmente necessárias para todos os domínios. Corresponde ao nível de execução de uma actividade de resolução de problemas levada a cabo por um STI.
- Uma *Framework* orientada a objectos – como sendo uma micro arquitectura que proporciona um modelo incompleto para sistemas dentro de um domínio de aplicação, por exemplo, um sistema construído com o objectivo de ser ampliado e reutilizado. Esta *Framework* de objectos é uma parte da *Framework* proposta e é composta por: uma visão de alto nível da arquitectura, baseada na arquitectura dos componentes comuns a todos os STI; um conjunto de componentes de programação incompletos consistindo em classes abstractas de objectos e suas relações.

Tirando partido das características da *Framework* proposta é apenas necessário desenvolver o que é específico para um particular domínio de aplicação, mais precisamente, ampliar e personalizar alguns dos componentes da *Framework*, definir a representação do conhecimento e implementação para cada tipo de conhecimento, no domínio a considerar, evitando-se a preocupação de começar tudo do zero.

Outras *Frameworks* para o desenvolvimento de STI são propostas por investigadores, como exemplo cite-se Liegle e Woo que propõem uma *Framework* para desenvolvimento de STI adaptativos, permitindo instrução personalizada através do uso da Internet, recorrendo a STI adaptativos. Como referido, a adaptação pode ser realizada baseando-se no conteúdo, ou baseada na sequência de abordagem de conteúdos. O objectivo é fornecer aos alunos instrução personalizada, tendo por base o seu estilo de aprendizagem, o conhecimento anterior e o conhecimento actual, [Liegle, Woo, 00].

A investigação e pesquisa em STI tem sido de certa forma “atrasada” pela falta de arquitecturas de sistema modulares, componentes reutilizáveis e bases de conhecimento partilháveis, em suma, pelos problemas identificados anteriormente e de seguida sintetizados. Cada nova investigação ou desenvolvimento tipicamente começa do zero, sem uma fundação por onde começar, [Patel et al, 01].

As considerações anteriores levam às seguintes conclusões:

Custos elevados de desenvolvimento – os investigadores são forçados a desenhar a sua própria arquitectura do sistema, implementar todos os componentes, desenvolver as estratégias de representação de conhecimento e mecanismos de raciocínio, adquirir e codificar toda a informação relevante acerca do domínio e conhecimento instrucional. Enquanto uma ou mais destas tarefas podem de facto constituir focos de pesquisa, as outras são meras tarefas necessárias para construir infra-estruturas que permitam que o trabalho prossiga para o cerne do problema em estudo.

Falta de interoperabilidade – uma grande variedade de ferramentas de software existe no mercado, e em laboratórios, mas estas ferramentas não podem ser facilmente importadas e utilizadas num STI. A tendência actual é de ferramentas modulares, componentes de software que permitam inter operacionalidade, no entanto pouca coisa se tem feito para permitir que as arquitecturas de STI tirem partido destas tendências.

Requisitos de plataformas bastante restritivos – Os STI implementados, tipicamente são programas de grande dimensão, desenhados sob a assumpção de grandes recursos computacionais. Alguns podem requerer hardware específico e/ou estarem implementados em linguagens que limitam a entrega e portabilidade para certos tipos de plataformas. Dado que os computadores pessoais são cada vez mais frequentes e acessíveis, quem desenvolve STI deve

procurar responder com arquitecturas e ferramentas multi-plataforma, não tão exigentes a nível de recursos.

Dificuldades de partilha de materiais e “benchmarking” – o desenho de sistemas actuais utiliza formalismos de representação de conhecimento idiossincrático, arquitecturas de dados internos e esquemas de controlo de fluxo. Como resultado pode ser extremamente difícil determinar os pontos fortes e fracos de aproximações individuais de tutoria, identificar e reutilizar unidades funcionais em vários domínios, e partilhar materiais de instrução e estratégias de tutoria.

Custos elevados de manutenção – STI, especialmente os protótipos de laboratório, são desenhados e construídos como aplicações complexas e embebidas de Inteligência Artificial, com pouca atenção dispendida nos problemas de desenvolvimento/evolução ou manutenção. Consequentemente, como algumas falhas de desenho não são “cobertas”, ou novos requisitos funcionais identificados, grandes partes do sistema frequentemente são descartadas e novamente implementadas de raiz.

Para equacionar estes problemas, a pesquisa em STI devia centrar-se na reutilização, na partilha e interoperabilidade de componentes de um STI. O objectivo deveria ser o redireccionar para uma redução do esforço necessário para construir um sistema e torná-lo viável. Nesta dissertação é apresentada uma *Framework* que pretende “aliviar” as preocupações anteriores e enfatizar novas direcções e contribuições que devem ser consideradas na construção de STI de nova geração. Esta *Framework* foi inicialmente apresentada em, [Rodrigues et al, 05].

A *Framework* de Desenvolvimento de Sistemas Tutores Inteligentes (FD-STI), caracteriza-se por tentar incorporar no desenvolvimento de STI outras áreas que não apenas a Informática ou as ciências dos computadores, nomeadamente a Psicologia e as Ciências da Educação, através das Teorias de Aprendizagem, como se pode observar na Figura 11. Uma característica importante é a separação da área de conhecimento e actuação do STI, do próprio sistema, assim como as estratégias pedagógicas que podem ser utilizadas. No primeiro caso o objectivo é obter STI independentes do domínio de conhecimento no qual serão utilizados, ou seja, um STI capaz de ser eficiente em qualquer domínio de conhecimento, acedendo à base de conhecimento externa ao próprio sistema, que se encontra numa forma normalizada, para obter o conhecimento que irá tentar transmitir ao aluno. No segundo caso, o objectivo essencial é permitir ao STI que seleccione uma de entre as várias formas de ensino, ou seja, que se baseie

numa estratégia pedagógica para proporcionar instrução, tornando também o STI (especialmente o módulo tutor), independente da Teoria de Aprendizagem a utilizar. Desta forma, permite-se que o STI se tente adaptar às realidades do aluno, à sua forma de aprender ou, como já foi referido, que se comporte como o professor ideal que muda de estratégias, personalidade, conhecimentos e forma de actuação perante situações e alunos diferentes.

Na FD-STI são também identificadas áreas de investigação de Informática, que se julgam importantes para o desenvolvimento de STI. Sugere-se que algumas das áreas de investigação, nomeadamente as suas contribuições, sejam validadas pela Psicologia Cognitiva e pelas Teorias de Aprendizagem, como forma de envolver investigadores de outras áreas e não apenas de Informática, no desenvolvimento de STI. Este envolvimento é referido como necessário e imprescindível por vários investigadores, sendo mesmo apontado por alguns como um factor de insucesso no caso em que esse envolvimento não se verifica. Poderão existir contribuições da Informática que não serão “validadas” ou analisadas em conjunto pelas Ciências da Educação, precisamente por não se relacionarem com aspectos pedagógicos da instrução que se pretende que o sistema proporcione. O objectivo é pois envolver investigadores provenientes de várias áreas, reflectindo a natureza multidisciplinar dos STI (Figura 1), não esquecendo que se está a desenvolver um sistema para aplicar numa área que não é determinística, desde logo porque lida com pessoas e com a forma como estas adquirem novo conhecimento.

Nesta visão, STI são sistemas baseados em conhecimento, que incorporam um ou mais dos seguintes esforços e características:

Componentes reutilizáveis – descrições de componentes reutilizáveis susceptíveis de serem utilizados em STI (ferramentas de modelação de utilizadores, por exemplo) que possam ser disponibilizados para a comunidade.

Esforços de normalização – que permitam a interoperabilidade de STI a diversos níveis, por exemplo ao nível de troca de conteúdos e partilha de modelos.

Vocabulários partilhados – trabalho ao nível de desenvolvimento de vocabulários partilhados, de parâmetros descrevendo as características dos alunos e o controlo táctico de tutoria (e.g. selecção de problemas, geração de explicações, tácticas de feedback).

Ontologias – ontologias estruturadas ou “upper models”, que definam e organizem atributos pedagogicamente relevantes de conhecimento para determinados domínios, permitindo a escrita e partilha de estratégias de instrução, em termos desses atributos.

Arquitecturas distribuídas e de agentes – arquitecturas e protocolos que envolvam processos de colaboração ou bases de conhecimento partilhado, que foquem aspectos de modularidade e reutilização. Estas incluem agentes autónomos e metodologias cliente/servidor, arquitecturas baseadas na Web, e integração de pacotes comerciais dentro dos STI.

Técnicas de personalização – técnicas já implementadas e testadas como modelação de utilizadores, gestão de conteúdos, publicação web, web mining. Aplicando métodos estatísticos e processos de data mining, é possível encontrar padrões interessantes que identificam o comportamento do aluno no sistema e que podem contribuir para uma melhor instrução.

Raciocínio baseado em casos – para armazenar as interacções dos alunos e o feedback com o STI, para que seja possível manter experiências passadas e derivar a partir delas novos caminhos de ensino.

Multimédia adaptativa – para seleccionar e apresentar conteúdos aos alunos numa dada plataforma, de tal maneira que responda às necessidades do aluno de acordo com o seu perfil, interacções passadas e motivações, e não apenas de uma forma pré determinada.

A participação da psicologia e das ciências da educação como referido é um requisito inevitável se pretendemos desenvolver STI que sejam eficazes e eficientes. Isto pode ser conseguido através de uma interacção com a Informática, nomeadamente através da validação das estratégias de tutoria a serem utilizadas pelo módulo tutor e através da escolha correcta da forma de apresentar os conteúdos ao aluno, de acordo com o modelo pedagógico mais adequado.

A Figura 11 ilustra a Framework de Desenvolvimento de Sistemas Tutores Inteligentes (FD-STI), apresentando-se de seguida algumas questões pertinentes relativas ao desenvolvimento de STI.

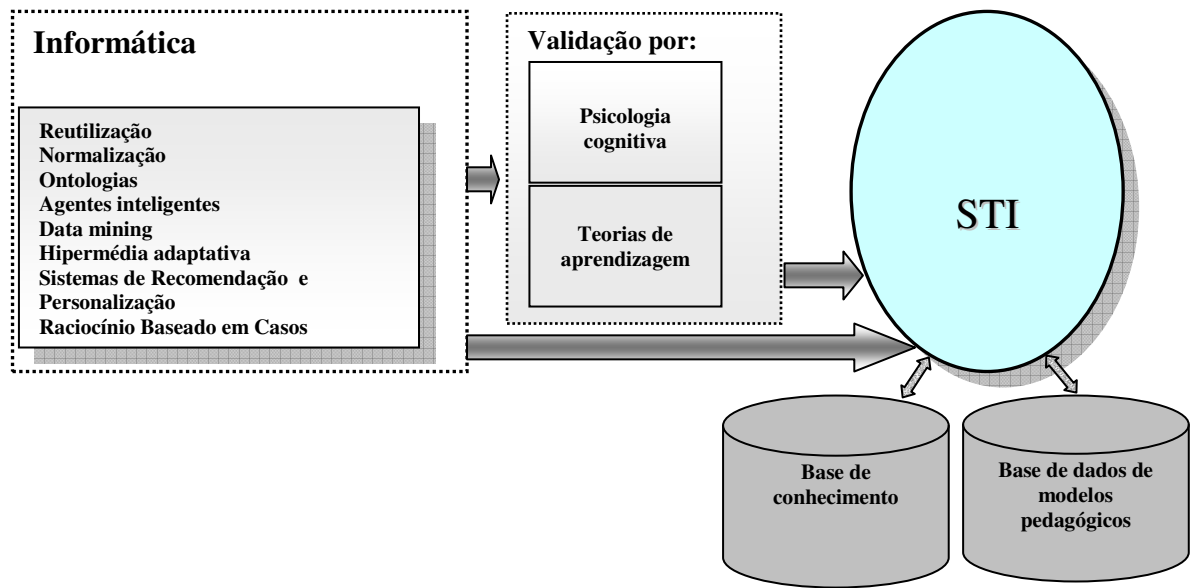


Figura 11 – FD-STI – Framework de Desenvolvimento de Sistemas Tutores Inteligentes

A FD-STI pretende dar resposta a algumas questões que constituem problemas pertinentes no desenvolvimento de STI, e que de seguida se enunciam:

- P1- Desenvolver STI é uma tarefa complexa;
- P2- Desenvolver STI envolve vários domínios e disciplinas;
- P3- Tipicamente, cada aplicação é desenvolvida independentemente, de raiz;
- P4- A “experiência do tutor é embebida no código do sistema;
- P5- Pouco envolvimento das ciências da educação e da psicologia;
- P6- Pouca reutilização de componentes como por exemplo a módulo tutor e o módulo interface;
- P7- Inexistência de linguagem normalizada para representação do conhecimento;
- P8- Dificuldades de comunicação entre componentes;
- P9- Falta de arquitecturas modulares;
- P10- Custos elevados de desenvolvimento;
- P11- Tempo de desenvolvimento;
- P12- Falta de interoperabilidade;
- P13- Requisitos das plataformas bastante restritivos;
- P14- Dificuldades de partilha de materiais/conteúdos;
- P15- Custos elevados de manutenção;

P16- Dificuldades de adaptação a alunos distintos.

Como forma de contribuir para a possível resolução destas questões, a FD-STI proposta visa propor técnicas que se julgam importantes para o desenvolvimento de STI, contribuindo com algumas melhorias. A Tabela 6 ilustra a Matriz problemas/contribuições, onde se procura sintetizar e referenciar a importância de cada uma das técnicas e metodologias, bem como as suas contribuições para uma tentativa de solucionar os problemas identificados, ou seja, para cada problema referido, a indicação de um “X” procura ilustrar que essa técnica/métodos tem um papel importante na resolução desse mesmo problema.

Tabela 6 - Matriz Problemas/contribuições

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16
Reutilização	X		X	X		X		X	X	X	X		X		X	
Normalização	X					X	X	X	X	X	X	X		X	X	
Ontologias		X			X		X	X	X			X				
Agentes Inteligentes	X		X	X	X	X		X	X	X	X				X	
Data Mining		X		X						X			X	X		X
Hipermédia Adaptativa		X		X	X	X				X	X			X	X	X
Sistemas de recomendação e personalização		X	X	X	X							X		X		X
Raciocínio Baseado em Casos	X	X	X	X						X	X			X		X

Nos sub – capítulos seguintes serão abordados os componentes da *Framework* e as questões apresentadas na matriz problemas/contribuições.

2.10. Reutilização

A reutilização de software foi definida como o processo de criar novas aplicações utilizando software ou pedaços de software previamente desenvolvidos, por contrapartida ao seu desenvolvimento a partir do zero. A reutilização é uma característica importante de um componente de software de qualidade, e presentemente, a visão de reutilização foi ampliada para incluir não apenas os algoritmos, mas também as estruturas de dados, [Arruarte et al, 96], [Karlsson, 95], [Pressman, 95].

Analogamente ao processo de desenvolvimento de software, o desenvolvimento de STI pode ser aperfeiçoado através da reutilização dos resultados de projectos anteriores. A reutilização pode-se referir a material de aprendizagem multimédia não produzido especificamente para o sistema em desenvolvimento ou, englobar a reutilização de partes de produtos das fases de projecto do sistema. O principal objectivo desta técnica é melhorar tanto a qualidade como a produtividade do software, [Marcke, Sarti, 95].

De um modo geral a reutilização pode ser decomposta em duas actividades, [Karlsson,95]:

- Desenvolvimento para reutilizar;
- Desenvolvimento com reutilização.

O desenvolvimento para reutilizar consiste numa actividade planeada, na qual partes de software são preparadas para que possam ser utilizadas noutros contextos. Essas partes reutilizáveis são denominadas de componentes reutilizáveis. Além disso, não estão restritas a nenhuma granularidade particular ou fase de ciclo de vida. Componentes de granularidade fina são geralmente genéricos e independentes do domínio. Fazem parte desta categoria componentes como funções de entrada/saída, funções de acesso a base de dados, funções de manipulação de estruturas de dados e classes individuais de objectos. Componentes de granularidade grossa podem ser subsistemas de aplicação tais como servidores de bases de dados, pacotes de interface com o utilizador, etc. O desenvolvimento para reutilizar estende o processo de desenvolvimento para as seguintes actividades:

- Analisar a variabilidade nas exigências entre diferentes utilizadores dos componentes;
- Analisar os custos e benefícios da incorporação dessas exigências;
- Projectar os componentes com o nível apropriado de generalidade para os utilizadores.

Além disso, o desenvolvimento para reutilizar inclui actividades de reengenharia de componentes existentes, qualificação e classificação dos componentes, para que estejam prontos para serem reutilizados.

O desenvolvimento com reutilização consiste na construção de novo software a partir de componentes reutilizáveis. Este processo inclui as seguintes actividades:

- Procurar um conjunto de componentes candidatos;
- Avaliar os componentes com o objectivo de se obter o mais apropriado;
- Se necessário, adaptar o componente seleccionado para que satisfaça as exigências.

O maior motivo pelo qual se tomam em consideração questões sobre a reutilização na área dos STI é devido ao facto de que a construção de STI requer ambientes de desenvolvimento razoavelmente grandes e dispendiosos. Levando-se em consideração que os sistemas construídos incrementalmente através de acréscimos e refinamentos sucessivos, o tempo e o custo de desenvolvimento são largamente reduzidos com a reutilização de software previamente desenvolvido.

A reutilização pode ser considerada também em relação ao conhecimento objecto de ensino, porém encontra dificuldades devido à falta de uma linguagem padrão para representar o conhecimento, de um interface padrão para permitir que as aplicações acessem ao conhecimento, e de um conjunto de ferramentas para permitir que os investigadores manipulem o conhecimento.

Para que a reutilização seja eficiente, os materiais a serem produzidos devem ser de fácil acesso, de fácil modificação e integração em contextos diferentes dos originais. Caso contrario, o custo de reutilização pode ser mais alto que o custo de desenvolver a partir do zero. Tornar os materiais facilmente acessíveis é a primeira condição necessária para se obter reusabilidade.

A base para se construir software de STI reutilizável é mantida pelo projecto modular dos seus componentes. O projecto modular permite não somente a reutilização como também a partilha e portabilidade. Partilha no sentido de que o mesmo conhecimento possa ser utilizado por vários sistemas implementados em diversas linguagens de programação; portabilidade no sentido em que o mesmo conhecimento possa ser instalado em plataformas diferentes.

Marcke e Sarti definiram reutilização da seguinte forma, [Marcke, Sarti, 95]:

Reutilização = restauração + adaptabilidade

Esta definição deu origem a regras de reutilização que embora pareçam óbvias e simples são difíceis de satisfazer na prática:

- Restauração: para seleccionar um componente a ser reutilizado, é necessário saber primeiro o que ele faz, para reutiliza-lo eficientemente, deve ser possível identifica-lo mais rapidamente do que se o fossemos construir;
- Adaptabilidade: para que uma técnica de reutilização seja eficiente, ela deve reduzir a distância cognitiva entre o conceito inicial do sistema de treino e a sua implementação final executável: além disso, deve ser mais fácil reutilizar os componentes ao invés de desenvolve-los a partir do zero, depois de um componente seleccionado, ele deve ser adaptado ao novo contexto.

A incapacidade da maioria dos softwares de aplicação educacional em partilhar dados padronizados com software de múltiplos fornecedores, e em trabalharem juntos para realizar objectivos em conjunto tem gerado discussões sobre a questão da compatibilidade. Na área da computação, essa compatibilidade entre múltiplos fornecedores é geralmente designada interoperabilidade

Aspectos como a interoperabilidade e a reutilização têm contribuído para aumentar a eficiência de ambientes baseados em computador em diversos domínios, e mostram-se promissores no aumento da eficiência nas investigações e desenvolvimento de STI. A interoperabilidade é muito mais que uma simples abordagem para se obter melhor custo - benefício e reutilização. A interoperabilidade facilita a criação de sistemas de grande porte através da definição de requisitos para transacções de interacção entre muitos sistemas menores, [Rowley, 96].

À medida que a interoperabilidade se tem tornado uma questão mais ampla na área da

computação, é importante considerar exemplos onde a interoperabilidade é bem sucedida. O exemplo mais comum é o da Internet. Neste caso, vários softwares cooperam para gerir múltiplas unidades de rede, cada uma ligada a uma espinha dorsal comum. Enquanto cada um dos softwares envolvidos numa única transacção na Internet pode ser sustentado por diferentes plataformas de hardware e fornecedores de sistemas de telecomunicações, cada transacção ocorre de forma confiável ao longo das várias redes. A interoperabilidade entre os componentes da Internet permite a criação de uma enorme infra-estrutura de telecomunicações, facilitando a proliferação de aplicações em áreas globais, [Rowley, 96].

Uma visão das possibilidades, dos níveis nos quais os padrões de interoperabilidade podem ser aplicados devem ser considerados assim como possíveis componentes de STI inter operáveis e requisitos para a implementação de padrões de interoperabilidade e reutilização entre STI, (por exemplo a normalização que será referida adiante). Esses padrões podem permitir vantagens como:

- Permitir que os STI tirem proveito de produtos comerciais compatíveis;
- Aumentar o custo – benefício e a reutilização dos componentes de STI em diversos domínios;
- Aumentar o nível de interacção entre STI relacionados.

Um ambiente ideal de pesquisa e desenvolvimento de STI inter operacionais deveria propor segundo, [Rowley, 96]:

- Possibilidade de criar STI personalizados a partir de componentes inter operacionais e reutilizáveis;
- Um padrão para definir um conjunto de protocolos de interface para módulos de sistemas de autor de STI, permitindo que no desenvolvimento se misturem e combinem componentes;
- Flexibilidade para que cada módulo do sistema inter operacional possa ser criado por investigadores ou laboratórios de pesquisa diferentes, cada qual seguindo os padrões de interface para comunicação entre módulos e de estruturação do conteúdo, bem como bases de dados de conhecimento para que possam ser utilizados entre todos os componentes inter operacionais de um STI;
- Possibilidades dos módulos componentes de um STI não dependerem de

conteúdos específicos, ou seja, não estarem limitados a esses conteúdos, tirando partido de recursos como a Internet para selecção e recuperação automática de recursos de aprendizagem inter operacionais para serem utilizados num STI central.

Esta abordagem de interoperabilidade para STI requer um alto nível de flexibilidade ao projectar os módulos do sistema. Exige cooperação e partilha de informação a alto nível entre os diversos investigadores, permitindo que muitos STI possam interagir e trabalhar juntos com o objectivo de alcançar objectivos maiores. Assim tornar-se ia possível criar novos STI com maior rapidez, mais eficientes, e independentes do domínio de aplicação.

2.11. Normalização

A normalização assume uma importância relevante nos dias de hoje, em qualquer área de conhecimento e de aplicação. Olhando à nossa volta podemos verificar que o mundo está cheio de exemplos de vantagens obtidas com a normalização. Desde as bitolas das linhas-férreas, ao formato das cassetes de áudio e vídeo, do encaixe das lâmpadas à Internet, às porcas e parafusos, entre muitos outros exemplos que têm permitido reduzir custos e simplificar a utilização. A não existência de componentes normalizados nos STI constitui pois um obstáculo relevante ao desenvolvimento de novos sistemas de forma mais rápida, fácil e económica. A normalização de componentes não significa necessariamente a normalização do conhecimento em geral. Pretende-se com a normalização, essencialmente fornecer um vocabulário comum para perceber o que já está feito até a data, com menos ambiguidade, não implicando qualquer restrição na exploração de actividades futuras de pesquisa.

A principal razão pela qual os humanos conseguem comunicar uns com os outros prende-se com o facto de existir uma plataforma comum de entendimento, na qual podemos confiar, e na existência de conceitos comuns com os quais podemos expressar as nossas ideias. Podemos pois referir que a normalização, pelo menos uma tentativa de normalização, para que se possa descobrir não apenas o que se pode normalizar, mas também o que não se pode, é crucial para o sucesso da investigação em STI. Na impossibilidade de uma normalização, poderemos recorrer a “vocabulários partilhados”, sendo que a especificação de componentes funcionais, poderia ser descrita em termos de um vocabulário comum. O problema consiste no facto de a terminologia utilizada por professores, autores e investigadores diferir de uns para os outros. Em resumo,

nenhum dos quatro intervenientes, três humanos e um computador, partilham o mesmo vocabulário, o que gera questões de desentendimento. Consequentemente, quando se tenta a comparação ou discussão entre diferentes STI relativos a domínios diferentes, torna-se numa tarefa difícil devido às diferentes terminologias utilizadas nos respectivos domínios. O mesmo se aplica à comunicação entre componentes modulares num STI. Devido à não existência de especificações funcionais comuns, ou de vocabulários comuns, não existe uma comunicação entre eles, sendo este um dos principais factores que impede a sua reutilização, [Mizoguchi, Bourdeau, 00].

Têm surgido cada vez mais esforços, na tentativa de encontrar standards para STI, tentando transpor para esta área todas as vantagens sobejamente conhecidas da sua aplicação noutras áreas de conhecimento. Todo este empenho prende-se com as vantagens inerentes à normalização como sejam:

- **Acessibilidade:** ao utilizar sistemas e conteúdos normalizados torna-se mais fácil o acesso aos mesmos, em qualquer localização, através de um browser;
- **Interoperabilidade:** os conteúdos são independentes das ferramentas que são utilizadas para lhes aceder, permitindo, por exemplo que diferentes STI acedam ao mesmo conteúdo, independentemente da ferramenta utilizada para o criar;
- **Adaptabilidade:** a normalização torna mais fácil proporcionar uma aprendizagem personalizada. Os metadados permitem descrever os conteúdos de forma a ser possível determinar o que o aluno necessita;
- **Reutilização:** tal como em qualquer área apenas é possível a reutilização de conteúdos se estes forem desenvolvidos de uma forma normalizada;
- **Durabilidade:** a normalização permite que, mesmo que a tecnologia de base mude, ainda seja possível utilizar o conteúdo sem a necessidade de o redesenhar ou recodificar;
- **Acessibilidade:** se os sistemas e conteúdos forem produzidos de forma normalizada a aprendizagem efectiva melhora assim como diminuem os custos e tempo de desenvolvimento.

Diversas instituições desenvolveram e desenvolvem iniciativas com vista à normalização, procurando standards utilizáveis por todos de forma a conseguir interoperabilidade de soluções de e-learning, facilitando também a interoperabilidade e o desenvolvimento de STI:

- Advanced Distributed Learning Network;
- IMS Global Learning Consortium;
- Aviation Industry CBT Committee;
- ARIADNE Alliance of Remote Instructional Authoring and Distribution Networks for Europe;
- IEEE Learning Technology Standards Committee;
- PROMETEUS.

Das normas definidas pelas organizações anteriores, uma delas merece particular destaque pela sua importância para os STI: a norma SCORM - Sharable Content Reference Model. É o modelo mais adoptado no mundo pela comunidade de e-learning, em grande parte por oferecer uma solução fiável para a implementação de sistemas inter operativos. Trata-se de um conjunto de especificações que têm sido aceites como norma para a produção de conteúdos. É descrito como uma prateleira contendo vários livros para diferentes propósitos, mas todos com o objectivo da criação e partilha de conteúdos de aprendizagem reutilizáveis. Esta biblioteca resultou da colaboração entre a ADL e IMS, define em XML, um formato para a estrutura de cursos, por forma a que os conteúdos educacionais possam ser reutilizáveis entre plataformas e sistemas. A Web é assumida como o principal meio para divulgação de cursos de aprendizagem, eliminando a anterior necessidade de adaptação a novas tecnologias.

Esta biblioteca está dividida em três áreas principais:

- Overview: Contém informação conceptual geral de introdução aos elementos do SCORM;
- Content Aggregation Model (CAM): define como empacotar os conteúdos de aprendizagem por forma a que possam ser reutilizados;
- Run-Time Environment (RTE): faz o interface entre o sistema e o aluno sendo independente das ferramentas utilizadas para a produção do conteúdo;

- Sequencing and Navigation (SN): permite representar sequências complexas e navegação de conteúdos de uma forma standard.

Esta norma assume um conjunto mínimo de serviços, tais como a abertura de um conteúdo, registo do progresso de cada aluno, gestão da ordem como cada objecto deve ser apresentado e relatório da progressão dos alunos. Note-se que num ambiente Web clássico, a interligação de objectos é realizada com hiperligações, mas na metodologia SCORM, o sistema define o que deve ser fornecido de seguida ao aluno, em função do que realizou no objecto anterior. Exemplo: se alcançou os objectivos mínimos estabelecidos, ser-lhe-á apresentada a próxima lição, caso contrário, eventualmente um conteúdo de recuperação. Este facto assume grande importância para o desenvolvimento de STI, pois, como já foi referido anteriormente, a necessidade de adaptação a cada aluno em particular é uma característica fundamental para um STI. Adicionalmente, a normalização de conteúdos permite aos investigadores focar a sua pesquisa no desenvolvimento dos vários módulos de um STI, deixando de ter preocupações relativamente à estruturação de conteúdos e à forma como o sistema os manipula, pois possuem a garantia que, sendo respeitadas as especificações o sistema conseguirá manipular os mesmos. Isto permite também um passo interessante relativamente à necessária independência do STI relativamente ao seu domínio de aplicação, caminhando-se no sentido de libertar a pesquisa e desenvolvimento de STI das questões relativas ao domínio de aplicação e aos conteúdos, focando-se noutras áreas como o desenvolvimento modular de componentes para STI por exemplo. O desafio reside na tentativa de definir normas e especificações para STI, analogamente ao que foi e está a ser feito com a norma SCORM para o desenvolvimento de conteúdos, ou seja, a possibilidade de desenvolver STI independentemente do domínio de aplicação, das linguagens e ferramentas utilizadas, desenvolvendo-se de forma modular, por equipas diferentes, existindo a garantia de perfeita integração de todos os componentes e da sua posterior reutilização se necessário.

2.12. Ontologias

O termo ontologia teve a sua origem na disciplina de Filosofia, onde designa uma explicação sistemática da existência. Definiu-se uma ontologia como sendo a explicação explícita de uma conceptualização. A ontologia fornece por outras palavras o vocabulário de representação para o domínio em questão, e um conjunto de definições e axiomas que restringem o significado dos termos nesse vocabulário, de forma a permitir uma interpretação consistente e única. A adesão a

uma ontologia comum garante a consistência (a mesma expressão tem o mesmo significado) e a compatibilidade (qualquer conceito é designado pela mesma expressão) da informação presente no sistema, [Gruber, 93].

De facto, a falta de uma ontologia comum para STI tem sido identificada por muitos investigadores como um factor limitativo, uma das “peças que faltam”. Enquanto que têm sido desenvolvidas ontologias para áreas de aplicação específica, não têm sido generalizadas para os domínios de aplicação de treino e ensino. Dada a diversidade de especificações dos conceitos relacionados com os STI, o desenvolvimento sistemático de ontologias formais deve pois ser prioritário, e os resultados deste esforço amplamente disseminado. Um tal esforço servirá para focar a atenção nesta “peça que falta”, e gerar as necessárias discussões dentro da comunidade de investigadores da área de STI procurando atingir um grau aceitável de consenso relativamente à ontologia proposta.

É pois fundamental a definição de ontologias na área de STI, como forma de universalizar a linguagem utilizada, facilitando aspectos anteriormente referidos como a reutilização, interoperabilidade e a normalização.

2.13. Agentes de Software

A tecnologia de agentes tem vindo a assumir um papel fundamental no desenvolvimento de STI, sendo vista por vários investigadores como a tecnologia a utilizar, e a que proporciona resultados mais interessantes. Pensando em ambientes de ensino distribuídos, recorrendo à Internet por exemplo, a utilização de sociedades baseadas em agentes está a consolidar-se como uma alternativa apropriada e muito interessante para o desenvolvimento de STI, [Bond, 90], [Frasson, 97], [Giraffa, Viccari, 99].

Um agente assume-se como uma figura metafórica utilizada em diversas áreas do conhecimento humano, que vão desde a Psicologia, a Economia e a Sociologia, até à Informática. Nesta última, o conceito de agente foi introduzido pela mão da comunidade da IA, em especial por intermédio do sistema Actor. Foram apresentados, pela primeira vez, novas entidades que foram designadas como objectos autocontidos, interactivos e concorrentes, que comunicavam por troca de mensagens e que demonstravam possuir um certo tipo de comportamento. Designaram-se esses objectos por actores, [Hewitt et al, 73].

Uma definição de agente passa por o considerar como algo que obtém informação e/ou conhecimento do seu universo de discurso através de sensores e actua nesse ambiente através de actuadores, como exemplificado na Figura 12. Pode-se dizer que os agentes são componentes persistentes e activos que percebem, raciocinam, actuam e comunicam. Pode-se perceber que os agentes são entidades que habitam em ambientes complexos, sentem esse ambiente e actuam de modo autónomo, procurando executar um conjunto de tarefas para as quais receberam procuração, [Russell, Norvig, 03], [Maes, 90].



Figura 12 - Agente a interactuar com o meio

Wooldridge e Jennings apresentam aquilo a que nos dias de hoje se pode definir como o fio condutor que levou a uma das primeiras formulações e/ou quantificações, sob um ponto de vista formal, da noção de agente ou Sistema Multiagente, [Wooldridge, Jennings, 95], [Wooldridge, 99].

Em particular, um agente ou Sistema Multiagente é considerado fraco quando não apresenta um conjunto mínimo de atributos, a partir dos quais se possam definir e quantificar um certo conjunto de atitudes como, por exemplo a autonomia, a reactividade, a pró-actividade e a sociabilidade.

Por outro lado, um agente ou Sistema Multiagente é considerado forte, quando a(s) entidade(s) com que nos deparámos é (são) eminentemente cognitiva(s), passível(eis) de desenvolver a sua própria consciência, de se apresentar como tendo um conjunto de mais valias como a perceptibilidade, a sentimentalidade e o emocionismo. Por conseguinte, um conjunto mínimo de atributos tem, de ser considerado, e em que se destacam, a mobilidade, a intencionalidade, a aprendizagem, a competência, a veracidade, a racionalidade, a benevolência, e a emotividade, [Russell, Norvig, 03], [Wooldridge, Jennings, 95], [Rao, George, 95].

Existem diversos tipos de agentes, podendo ser: de software ou de hardware, estacionários ou móveis, persistentes ou temporários, reactivos ou cognitivos.

A abordagem centrada em agentes preocupa-se em estudar o comportamento de uma sociedade constituída por agentes, que possuem autonomia, e têm como objectivo realizar tarefas que não são possíveis de serem realizadas individualmente.

A justificação da aplicação da tecnologia de agentes na concepção de Sistemas de Informação é justificada quando o problema possui as seguintes características [Jennings et al, 96]:

- O domínio envolve distribuição intrínseca dos dados, capacidade de resolução de problemas e responsabilidades;
- Necessidade de manter a autonomia de sub partes, sem a perda da estrutura organizacional;
- Complexidade nas interacções, incluindo negociação, partilha de informação e coordenação;
- Impossibilidade de descrição da solução do problema *a priori*, devido à possibilidade de perturbações em tempo real no ambiente e processos de negócio de natureza dinâmica.

Possuindo o software educativo as características referidas, a tecnologia de agentes assume pois um papel importante no seu desenvolvimento. As propostas de utilização de arquitecturas SMA em STI trazem uma grande vantagem em relação às arquitecturas tradicionais, por apresentarem uma maior flexibilidade no tratamento dos elementos que compõem o sistema. Além disso, o facto de utilizar agentes para modelar os seus componentes possibilita o agrupamento da arquitectura tradicional (um módulo = um agente) ou a explosão de cada módulo em vários agentes. A modelação de STI numa arquitectura funcional de agentes é mais do que uma abordagem generalista. Teoricamente a modelação não apresenta limites para o número de agentes que podem participar do processo de aquisição do conhecimento. Processo este que ocorre através de negociação dos papéis dos agentes, tanto tutores, como aprendizes.

Frasson refere que “Uma das mais promissoras aplicações de agentes autónomos está provavelmente na área de educação e treino,” [Frasson, 97].

Existem diversos exemplos na literatura sobre a utilização de agentes em sistemas educacionais. Uma sociedade de agentes para aprender e ensinar, pode ser a solução para a construção de ambientes de ensino e aprendizagem, se os agentes trabalharem de uma maneira concorrente e autónoma para alcançar os seus objectivos. Os agentes num ambiente de ensino - aprendizagem são considerados autónomos porque: as actividades dos agentes individuais não requerem constante supervisão externa (humana), e não há uma autoridade central projectada para controlar todas as interacções desempenhadas entre os agentes.

O uso de agentes na concepção de sistemas educacionais traz algumas vantagens, tais como: reagir às acções do utilizador, credibilidade, modelação de sistemas colaborativos multi-utilizador e modularidade, pelo facto de que cada agente é um módulo único e independente do outro sendo mais fácil adicionar outros agentes a estes sistemas. As vantagens são as seguintes:

- O conhecimento pode ser distribuído entre vários “tutores”, cada um com as suas crenças, desejos, objectivos, emoções e planos de acção. Esta distribuição cria maiores oportunidades de variar as técnicas pedagógicas utilizadas;
- O aluno interage com um tutor de forma mais flexível;
- O aluno pode passar conhecimentos ao tutor que serão retransmitidos a outros alunos.

As tecnologias baseadas na Web em conjunto com metodologias multiagentes formam uma nova tendência na modelação e desenvolvimento de ambientes de aprendizagem. As metodologias multiagentes tem surgido com uma alternativa para conceber aplicações de aprendizagem distribuída, devido ao conjunto de características inerentes ao conceito de SMA e às peculiaridades de uma sociedade de agentes. A principal razão deve-se ao facto que esta tecnologia lida muito bem com aplicações críticas, tais como: distância, cooperação entre diferentes entidades e integração de diferentes componentes de software.

No futuro, segundo alguns investigadores, os ambientes da aprendizagem estarão acessíveis a qualquer lugar e a qualquer hora. Os alunos desses ambientes estarão distribuídos no espaço e no tempo. Logo, trabalhos que utilizam arquitecturas multiagentes oferecem uma promissora abordagem para o projecto desses ambientes. Com a modularidade e a uniformidade dos agentes e com a padronização dos protocolos de interacção, o nível de escalabilidade e interoperação pode ser alcançado, o que não pode ser conseguido tão facilmente com o uso de outras técnicas.

As arquitecturas multiagentes permitem o constante crescimento e a heterogeneidade do ambiente de software.

No entanto a abordagem do desenvolvimento de STI através de agentes enfrenta alguns problemas:

- A complexidade em modelar e implementar STI e ambientes de ensino-aprendizagem inteligentes. A sua arquitectura é mais complexa e a modelação dos seus componentes e inter relações é demorada, necessitando de uma grande quantidade de trabalho cooperativo recorrendo a equipas interdisciplinares;
- A tecnologia de agentes aplicada a tais ambientes agrega mais complexidade ao seu projecto.

No entanto, os avanços mais recentes no campo dos ambientes de aprendizagem inteligentes, têm proposto o uso de arquitecturas baseadas em sociedades de agentes. Os princípios dos sistemas multiagentes têm mostrado um potencial bastante adequado ao desenvolvimento de sistemas de ensino, devido ao facto de a natureza do problema de ensino-aprendizagem ser mais facilmente resolvido de forma cooperativa. Além disso, ambientes de ensino baseados em arquitecturas multiagentes possibilitam suportar o desenvolvimento de sistemas de forma mais robusta, mais rápida e com menores custos, [Villareal, Giraffa, 01].

Os agentes desenvolvidos para ambientes de ensino recebem o nome de agentes pedagógicos. Os seus objectivos podem ser descritos em função do seu comportamento (estratégia). Os comportamentos possíveis para um tutor são:

- Guia: o agente é directivo nas suas intervenções e monitoriza o aluno, conduzindo-o na resolução do problema por todo o processo de interacção;
- Assistente: o agente é menos directivo mas também monitoriza o aluno, intervindo baseando-se em heurísticas sobre a resolução do problema naquele domínio;
- Facilitador: o agente monitoriza o aluno porém não é directivo, ele apenas dá dicas sobre a resolução do problema e só intervém quando solicitado.

O tutor pode utilizar um ou vários comportamentos, sendo assim, os objectivos podem variar em função das informações recebidas do utilizador (acções do aluno) e da situação que se deseja atingir (plano).

Os agentes pedagógicos podem actuar como tutores virtuais, alunos virtuais ou colegas virtuais que auxiliam no processo de aprendizagem. Estas diferentes formas de actuação podem compor grupos de agentes que distribuem entre si as tarefas. A distribuição de tarefas permite um melhor desempenho do sistema, subdividindo a complexidade da tutoria em tarefas menores. A tarefa de tutoria passa a ser executada num nível mais alto de abstracção. Desta forma, alguns investigadores sugerem uma caracterização do processo de aprendizagem desenvolvido em sistemas multiagentes em três níveis de abstracção, [Gouarderes et al, 99]:

- Aprendizagem por replicação, onde os agentes disponibilizam o conteúdo, a representação da estratégia pedagógica seleccionada, e dentro desta arquitectura um dos agentes representa as acções do professor, e o processo de aprendizagem está representado de forma reactiva pelos agentes;
- Aprendizagem por tautologia, onde a apresentação do conteúdo é desenvolvida de forma a guiar o aluno através de agentes especializados que auxiliam o aluno;
- Aprendizagem por interacções dinâmicas e partilhadas, onde o computador é mais activo e a informação não é fornecida somente pelo sistema, mas pode ser modificada e/ou gerada pelo aluno, desta forma o processo de aprendizagem caracteriza um modelo cognitivo.

Vários sistemas foram desenvolvidos utilizando a tecnologia de agentes, destacando-se:

- **White Rabbit** – desenvolvido pelo Departamento de Informática e Pesquisa Operacional da Universidade de Montreal - Canadá, este sistema tem como objectivo aumentar a cooperação entre um grupo de pessoas pela análise dos seus diálogos. Cada utilizador é assistido por um agente inteligente o qual estabelece um perfil dos seus interesses. Com o comportamento móvel e autónomo o agente pesquisa agentes pessoais de outros utilizadores, com o objectivo de encontrar aqueles que tenham interesses comuns para os colocar em contacto, [Thibodeau et al, 00];
- **Lecs** – O LeCS (Learning from Case Studies) foi desenvolvido pelo Departamento de Computação e Estatística da UFSC – Brasil, em conjunto com a Unidade de Aprendizagem Baseada em Computador da Universidade de Leeds – UK, LeCS é um sistema inteligente de aprendizagem à distância, que

possui uma arquitectura baseada num Sistema Federativo de agentes. A arquitectura inclui três classes de agentes: Agente Interface, Agente de Informação e Agente Conselheiro, [Rosatelli et al, 00];

- **Lanca** - No projecto LANCA, os autores procuram expor o porquê do uso de agentes inteligentes em STI e como podem ser adaptados para a aprendizagem à distância. O projecto foi desenvolvido pelo grupo do Departamento de Informática e Pesquisa Operacional da Universidade de Montreal – Canadá, e da Unidade de Informática da Universidade de Pau – França. LANCA apresenta as principais características dos agentes para um ambiente de aprendizagem à distância, bem como as suas funções em ambientes distribuídos. Propõem, também uma arquitectura para o ambiente com a especificação dos papéis dos diferentes agentes inteligentes que compõem a sociedade, [Frasson et al, 98];
- **Baguera** - O projecto Baguera, foi desenvolvido pela Universidade de Grenoble – França, cujo objectivo é desenvolver uma fundamentação teórica e metodológica para guiar a concepção e modelação de ambientes de aprendizagem. A plataforma Baguera está fundamentada no princípio de que a função educacional do sistema está nas interacções organizadas entre os componentes: agentes e humanos e, não meramente na funcionalidade de uma das suas partes. O primeiro resultado desse projecto inclui uma arquitectura multiagente baseada na Web para ambientes de aprendizagem e um protótipo para a aprendizagem de geometria, [Webber et al, 01];
- **I-Help** - O I-Help foi desenvolvido pela Universidade de Saskatchewan – Canadá, este projecto descreve uma infra-estrutura multiagente para o I-Help, um ambiente de aprendizagem baseado na Web para auxiliar alunos na solução de problemas. O sistema contém uma variedade de recursos da aprendizagem, fóruns, materiais *on-line*, *chat*, etc, [Vassileva et al, 01].
- **The Explanation Agent** - O Explanantion Agent ou Agente de Explicações, desenvolvido pelo Departamento de Informática e Pesquisa Operacional da Universidade de Montreal – Canadá, tem como objectivo principal prover respostas ou explicações sobre o conteúdo com maior qualidade, identificando problemas que possam ocorrer durante o processo de explicação ou resolução de problemas. Tem dois objectivos específicos: descobrir a fonte do mal

entendido do aluno através do modelo do aluno, e ajudar o “projectista” do curso a adaptar as suas explicações de acordo com estas observações. É utilizada a teoria de Mapas Conceptuais para estruturar as explicações numa representação formal. Esta representação é usada pelo Agente Explicação para efectuar as suas deduções sobre conceitos mal entendidos pelo aluno, [Zouaq et al, 00].

2.14. Data Mining

O crescimento exponencial da quantidade de informação armazenada, bem como a drástica redução dos custos de armazenamento dessa mesma informação tem contribuído para que nos últimos anos as organizações possuam à sua disposição uma enorme quantidade de dados, que abre novas oportunidades. Proporcionalmente a essa quantidade de dados, existe conhecimento neles armazenado, que pode ser extraído e derivado, identificando-se padrões e tendências, e tentando inferir regras para os mesmos. A análise de dados pode fornecer um conhecimento adicional, ao permitir ir além dos dados guardados explicitamente.

Esta informação preciosa está na verdade implicitamente escondida sob uma montanha de dados, e não pode ser descoberta utilizando os sistemas de gestão de bases de dados convencionais. Através do Data Mining podemos no entanto obter essa informação. Data Mining é uma tecnologia utilizada para revelar informação estratégica escondida em grandes volumes de dados. É utilizada em diversas áreas, como análise de riscos, marketing direccionado, controle de qualidade, análise de dados científicos, etc. Define o processo automatizado de captura e análise de enormes conjuntos de dados, para então extrair um significado. Esta tecnologia está a ser utilizada para descrever características do passado, assim como prever tendências para o futuro. A sua utilização permite avanços tecnológicos e descobertas científicas, além de garantir uma vantagem competitiva invejável, [Zytkow, Klosgen, 01].

Data Mining pode pois ser considerado como a aplicação de métodos e técnicas em grandes Bases de Dados, para encontrar tendências ou padrões com o intuito de descobrir conhecimento, [Santos, 03].

A área de Data Mining Educativo, é uma disciplina emergente, que se preocupa com o desenvolvimento de métodos para explorar os tipos de dados únicos, que se obtêm de aplicações

educacionais, e com a utilização desses métodos para melhor “perceber” os alunos, os locais onde aprendem e a forma como aprendem. Independentemente da informação (dados educativos), ser obtida através da utilização por parte dos alunos de ambientes educativos interactivos, ou dados administrativos de escolas e universidades, possui múltiplos níveis de hierarquias, que frequentemente necessitam de ser determinados através das propriedades dos próprios dados, em vez de serem determinados à priori. Itens como tempo, sequência e contexto também desempenham um papel importante no estudo de dados educativos [educationaldatamining].

O desenvolvimento de sistemas de e-learning, especialmente daqueles que utilizam como base a web, nos últimos anos, tem também motivado a aplicação de técnicas de Data Mining e descoberta de conhecimento como ferramentas para poder melhorar a aprendizagem em tais sistemas. Essas técnicas de Data Mining e descoberta de conhecimento têm sido aplicadas com êxito em sistemas de comércio electrónico para compreender o comportamento de clientes e poder incrementar as vendas. Ainda que as técnicas de descoberta de informação utilizadas em e-learning e e-commerce sejam similares, os objectivos finais têm matrizes totalmente diferentes, dado que no e-commerce o objectivo é guiar os clientes durante a compra para a maximizar, no e-learning o objectivo é guiar os alunos na sua aprendizagem, para a maximizar. As principais aplicações de técnicas de Data Mining em educação podem ser agrupadas em: sistemas de personalização; sistemas de recomendação; sistemas de modificação e sistemas de detecção de irregularidades, devido às suas capacidades para: descobrir padrões de navegação regulares e irregulares; classificação de alunos e de conteúdos; construção adaptativa de planos de ensino; descoberta de relações entre actividades; diagnostico incremental dos alunos etc., [Srivastava et al, 00], [Li, Zaiane, 04], [Perkowitz, Etzioni, 98], [Zaiane, 01].

A aplicação de técnicas de Data Mining em educação pode ver-se de dois pontos de vista distintos ou orientações:

- Orientado para os autores – com o objectivo de ajudar os professores e autores para que possam melhorar o funcionamento e/ou rendimento dos sistemas a partir da informação de utilização dos alunos. As suas principais aplicações são: obter uma maior re-alimentação do ensino, conhecer mais sobre como os alunos aprendem, avaliar os alunos pelos seus padrões de navegação, reestruturar conteúdos para personalizar cursos, classificar alunos em grupos, etc;

- Orientado para os alunos – com o objectivo de ajudar ou fornecer recomendações aos alunos durante a sua interacção com o sistema para poder melhorar a sua aprendizagem. As suas principais aplicações são: sugerir boas experiências de aprendizagem aos alunos, adaptação do curso segundo o progresso do aluno, ajudar os alunos fornecendo sugestões e atalhos, recomendar caminhos mais curtos e personalizados etc.

Como referido o Data Mining é uma área multidisciplinar onde convergem diferentes paradigmas de computação sendo os seus principais objectivos [Santos, 03]:

- Classificação – encontrar uma função que associa um caso a uma classe de diversas classes discretas de classificação, para classificar um novo objecto de acordo com o modelo de classificação;
- Previsão – prever valores futuros ou desconhecidos de outras variáveis de interesse, descoberta de padrões a partir de exemplos;
- Regressão – encontrar uma função para uma previsão de uma variável;
- Segmentação – identificar um conjunto finito de categorias ou segmentos para descrever os dados, identificar grupos homogéneos que constituem uma classe;
- Associação e dependência – encontrar modelos que descrevam dependências entre variáveis;
- Sumariação – encontrar descrições compactas para grupos de dados;
- Visualização – apresentação dos resultados finais ou intermédios, do Data Mining;
- Detecção de desvios - detectar alterações significativas nos dados, a partir de valores medidos ou padronizados.

O Data Mining é na realidade um dos passos que compõem o processo de descoberta de conhecimento composto por:

- Pré processamento – consiste na recolha e extracção de dados, selecção de atributos e integração dos dados;

- Data Mining – selecção dos algoritmos de Data Mining a utilizar e na aplicação dos algoritmos sobre os dados;
- Pós-processamento – consiste na interpretação e avaliação dos resultados obtidos e na utilização do conhecimento descoberto.

A aplicação de técnicas de Data Mining para extrair conhecimento de dados da web designa-se como Data Mining Web. Tendo presente que vários autores referem a Internet como palco preferencial para a utilização de STI, percebe-se a importância do Data Mining Web, que pode consistir essencialmente em três tipos:

- Data Mining de conteúdos – processo de extrair informação a partir dos conteúdos dos documentos web;
- Data Mining de estrutura web – processo de descobrir informação a partir da estrutura web;
- Data Mining de utilização web – processo de descobrir informação a partir dos dados de utilização da web.

Destes três tipos, o mais utilizado para a descoberta de informação em sistemas de ensino baseados na web é o Data Mining de utilização Web. No entanto vários trabalhos têm sido desenvolvidos, tendo os investigadores utilizado várias técnicas para atingir os seus objectivos sendo no entanto de realçar:

- A classificação e agrupamento – onde se pretende agrupar os utilizadores pelo seu comportamento de navegação, agrupar as páginas pelo seu conteúdo, tipo ou acesso, e agrupar os comportamentos de navegação similar, como referido em, [Srivastava et al, 00],[Li, Zaiane, 04],[Perkowitz, Etzioni, 98];
- Descoberta de regras de associação – permite descobrir relações ou associações entre diferentes páginas web visitadas, utilizando por exemplo agentes recomendadores, ou analisando a transferência de aprendizagem entre alunos, através da análise de ficheiros de logs de interacção de STI, como referido em, [Zaiane, 01],[Zaiane, 02],[Freyberger et al, 04];
- Detecção de padrões – detecção de padrões ou análise de sequências num conjunto de dados, permitindo por exemplo analisar sequências de páginas visitadas pelos alunos, numa ou em várias sessões. A análise de padrões de

comportamento permite descobrir os caminhos de aprendizagem ideais, dentro da topologia de um curso como referido em, [Becker et al, 04], [Freyberger et al, 04], [Wang et al, 04].

Uma outra técnica utilizada em sistemas de ensino é o Data Mining de textos. Tendo presente que cerca de 80% do conteúdo on-line está no formato de texto e que 80% das informações armazenadas por uma empresa são também dados não estruturados, o Data Mining de textos assume um papel preponderante também nos STI, [Tan, 99].

As técnicas de Data Mining servem também para monitorizar e avaliar a aprendizagem dos alunos em STI. Num ambiente de ensino tradicional, a observação do aluno é uma das técnicas informais mais usadas na detecção de falhas na aprendizagem. Logo, a interacção face-a-face entre o professor e o aluno, nesse tipo de ensino, constitui a principal fonte de informações desse método. Este é, também, o maior obstáculo para o acompanhamento da aprendizagem em STI, já que, na maior parte do tempo, o contacto com aluno é mediado pelo computador, limitando-se possivelmente às mensagens escritas. A grande questão hoje, é como superar as dificuldades impostas pela distância. A tecnologia tornou possível capturar, à distância, algumas características do aluno e analisá-las de uma maneira análoga ao comportamento de um aluno de um curso presencial. A linguagem corporal, o grau de interesse, a participação, o comportamento social, podem ser vistos pela óptica computacional, considerando, basicamente, as interacções do aluno com o ambiente de ensino. A frequência da sua participação em listas e chats, por exemplo, pode retratar a sua sociabilidade. Análises mais aprofundadas dos dados disponíveis permitem uma generalização do comportamento dos alunos, por meio dos padrões encontrados, sem deixar de lado as suas características individuais, relacionando os padrões mais adequados a cada aluno. Desse modo, a aplicação de Data Mining poderá ser útil na implementação de STI que permitam um maior acompanhamento dos alunos, [Silva et al, 01].

2.15. Hipermedia Adaptativa

Hipermedia Adaptativa (HA) é uma área oriunda da união da investigação em Hipermedia e Modelação de Utilizadores, visando obter sistemas capazes de responder de forma diferenciada a cada utilizador, conforme a sua necessidade de informação, baseando-se num modelo que represente as suas metas, o seu conhecimento e as suas preferências específicas. Desse modo, permite-se que o utilizador possa usufruir de uma forma mais adequada de navegação e

apresentação dos conteúdos do domínio em estudo, o que desde logo se enquadra perfeitamente no desenvolvimento de STI.

Na Figura 13 é apresentado o ciclo clássico de sistemas adaptativos, segundo Brusilovsky.

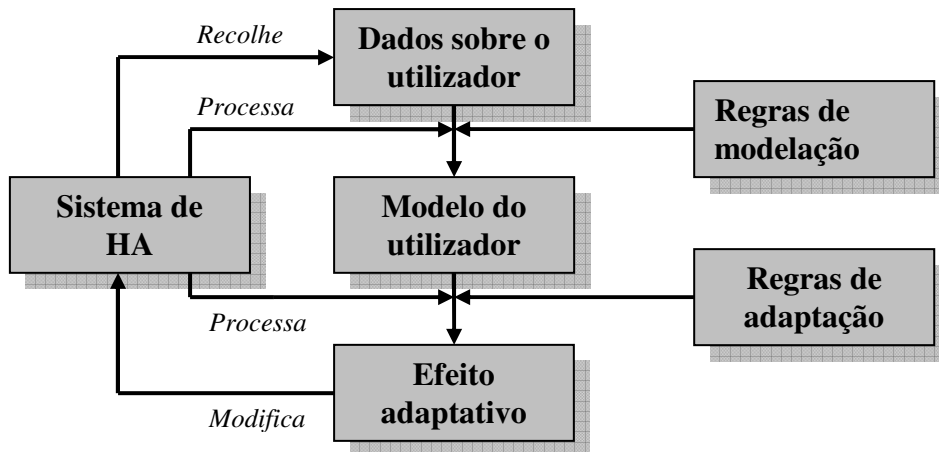


Figura 13 - Ciclo clássico dos sistemas adaptativos adaptado de [Brusilovsky, 96]

O ciclo de adaptação clássico, segundo Brusilovsky, consiste em duas etapas bem definidas: modelação do utilizador e adaptação, [Brusilovsky, 96].

A modelação do utilizador é a fase onde o sistema, após ter recolhido os dados do utilizador, os incorpora no seu modelo. Nessa fase ocorre o mapeamento dos eventos ocorridos na interface do sistema para um modelo que representa as características de cada utilizador.

Sempre que o modelo do utilizador é alterado, o sistema passa a considerar as alterações, visando estabelecer as adaptações devidas. A etapa de adaptação disponibiliza ao utilizador novas opções de interação com o sistema por meio de possíveis alterações na interface, representando o novo estado do modelo do utilizador.

Desse modo, forma-se um ciclo, onde a cada nova interação, outros processos de modelação e adaptação ocorrem, proporcionando uma nova interação com o sistema, prosseguindo continuamente até o fim da sessão de tutoria.

Alguns dos métodos e técnicas de HA foram descritos por Brusilovsky. Nesses trabalhos é apresentada uma classificação inicial quanto aos espaços de adaptação, [Brusilovsky, 96], [Brusilovsky, 99]:

- Apresentação Adaptativa
- Navegação Adaptativa.

A primeira trata da adaptação do conteúdo em si, permitindo que sejam apresentadas ao utilizador diferentes páginas, conforme as características registadas e modeladas. A segunda trata da adaptação dos *links*, ou seja, da variação das possibilidades de navegação.

Conforme a classificação feita por Brusilovsky, os métodos de Apresentação Adaptativa são, [Brusilovsky, 96]:

- Explicações Adicionais - o sistema KN-AHS utiliza este método, [Kobsa, 01];
- Explicações de Pré-requisitos - um exemplo de sistema que emprega este método é o LISP-CRITIC. Um sistema que permite que os programadores solicitem sugestões sobre como melhorar o seu código LISP de forma gradual, [Fisher et al, 90];
- Explicações Comparativas - o sistema C-Book fez uso desse método, [Kay, Kummerfeld, 94];
- Variantes de Explicação - De entre os sistemas que implementaram este método está o Anatom-Tutor, [Beaumont, 94];
- Classificação de Fragmentos de Informação - Este método foi utilizado pelo sistema EPIAIM, um interface inteligente para um pacote estatístico que auxilia pessoas com pouca experiência na análise correcta de dados sobre epidemiologia, [De Rosis et al, 94].

Segundo Brusilovsky, um sistema adaptativo no seu nível de implementação pode utilizar as seguintes técnicas de apresentação adaptativa, [Brusilovsky, 96]:

- Texto Condicional - esta técnica permite implementar todos os métodos anteriormente descritos, excepto o de Classificação de Fragmentos de Informação. Foi implementada, por exemplo, pelos sistemas ITEM/IP e C-BOOK anteriormente referidos;
- Texto Alongado - os métodos de Explicações Adicionais, Explicações de Pré-requisitos, e Explicações Comparativas também podem ser implementados por meio da técnica de Texto Alongado. O sistema KN-AHS empregou esta técnica por meio de palavras destacadas, conforme citado anteriormente;

- Variantes de Fragmento - um exemplo do seu uso é o sistema Anatom-Tutor que combina a técnica de Variantes de Página com a técnica de Variantes de Fragmento para montar a melhor combinação de fragmentos e páginas que atenda ao nível de conhecimento do utilizador;
- Variantes de Página - Essa técnica foi empregue, por exemplo, nos sistemas Anatom- Tutor e C-book, referidos anteriormente;
- Janelas - permite implementar todos os métodos de apresentação adaptativa. O sistema EPIAIM citado anteriormente utilizou janelas para representar os fragmentos de informação a serem classificados, [Hook et al, 96].

Além do aspecto de adaptação do conteúdo das páginas, é possível analisar-se outro aspecto: adaptação da navegação. Por meio de métodos e técnicas de navegação adaptativa os *links* que fazem parte da página exibida ao utilizador são dinamicamente seleccionados, conforme o modelo específico do utilizador. Esses métodos e técnicas foram classificados em, [Brusilovsky, 01]:

- Condução Global - tenta apresentar o caminho mais apropriado para chegar ao objectivo. Por exemplo, em Sistemas Educativos, tais como WebWatcher, Adaptive Hyperman e HYPERFLEX, o objectivo está relacionado com a aprendizagem dos conceitos do curso, [Armstrong et al,95], [Mathé, Chen, 94], [Kaplan et al, 93];
- Condução Local - este método é utilizado para indicar que conceitos podem ser alcançados a partir do conceito actual, como implementado nos sistemas Adaptive HyperMan e HYPERFLEX;
- Suporte à Orientação Local - o sistema ELM-ART implementa o método de Suporte à Orientação Local com o foco no conhecimento do utilizador, por meio da metáfora de um semáforo de trânsito, que indica *links* para páginas cujo conteúdo o utilizador ainda não está pronto para aprender, [Brusilovsky, 01];
- Suporte à Orientação Global - no sistema ISIS-Tutor de Brusilovsky os links que supostamente ainda não devem ser aprendidos pelo utilizador são simplesmente ocultados, visando simplificar tanto a navegação quanto a aprendizagem;

Os métodos de navegação adaptativa descritos anteriormente são implementados por meio das técnicas:

- Orientação Directa - consiste em apresentar ao utilizador qual a próxima página que deveria ser visitada, e.g. o sistema WebWatcher destaca visualmente o *link* para o "melhor" nodo, enquanto que o sistema ISIS-Tutor apresenta dinamicamente um *link* adicional para a próxima página recomendada
- Classificação Adaptativa - baseia-se na ideia de classificar os links de uma página em particular, conforme o modelo do utilizador, colocando os links mais valiosos no topo da lista, e os menos relevantes no final da lista de links.
- Ocultação de Links – consiste em restringir o espaço de navegação por meio da ocultação dos *links* não relevantes ;
- Anotação de Links - os *links* são acrescidos de comentários que visam informar o utilizador do tipo de informação que será exibida caso escolha determinado *link*;
- Adaptação de Mapas - compreende todas as formas de adaptar a navegação por meio de imagens que representam o domínio global ou local do sistema de hipermédia em questão.

A classificação de *links* segundo Brusilovsky é a seguinte, [Brusilovsky, 01]:

- Links locais não contextuais: São links independentes do conteúdo da página. Possuem a forma de botões, listas, ou menus pop-up. Estes links podem ser classificados, ocultos ou anotados.
- Links contextuais: Este tipo corresponde aos links que aparecem dentro de um texto ou de uma imagem, estando portanto, vinculados ao contexto e impossibilitados de serem classificados ou completamente ocultos. Normalmente, aplica-se a técnica de Anotação de Links.
- Links de páginas de índices e de tabelas de conteúdos: Estes links compõem um tipo especial de página que contém apenas links. Quando os links estão ordenados alfabeticamente a página é denominada página de índices, e quando os links aparecem ordenados conforme o conteúdo das páginas, a página de

links recebe o nome de tabela de conteúdo. Estes links são normalmente não contextuais, excepto quando implementados por meio de uma imagem.

- Links em mapas locais e globais: Mapas são representações gráficas do espaço global ou local de navegação, consistindo normalmente em nodos interligados por setas. Ao seleccionar um dos nodos, o utilizador tem acesso directo à página correspondente ao nodo.

Esta classificação das técnicas de Hipermedia Adaptativa não é estática, devido à contínua evolução das tecnologias de interacção entre o homem e o computador. A Figura 14 mostra a revisão dessa classificação feita por Brusilovsky, onde se pode observar o surgir de novas classificações, [Brusilovsky, 01]:

- Multimédia Adaptativa: técnicas que manipulam não apenas textos, mas figuras, áudio e vídeo, entre outros objectos multimédia;
- Adaptação da Modalidade: técnicas que se preocupam com a possibilidade do utilizador utilizar diferentes dispositivos para aceder ao mesmo sistema, como por exemplo, desktops, hand-helds e telemóveis (m-learning por exemplo);
- Adaptação da Linguagem Natural: técnicas que adaptam frases fornecidas pelo sistema ao utilizador, gerando frases sintáctica e semanticamente correctas, utilizando técnicas de linguagem natural.

Brusilovsky divide a HA nas seguintes áreas de aplicação:

- Hipermedia educativa, sistemas de informação on-line, sistemas de ajuda on-line, hipermedia de recuperação de informações;
- Hipermedia institucional, e sistemas para gerir visões personalizadas em espaços de informação.

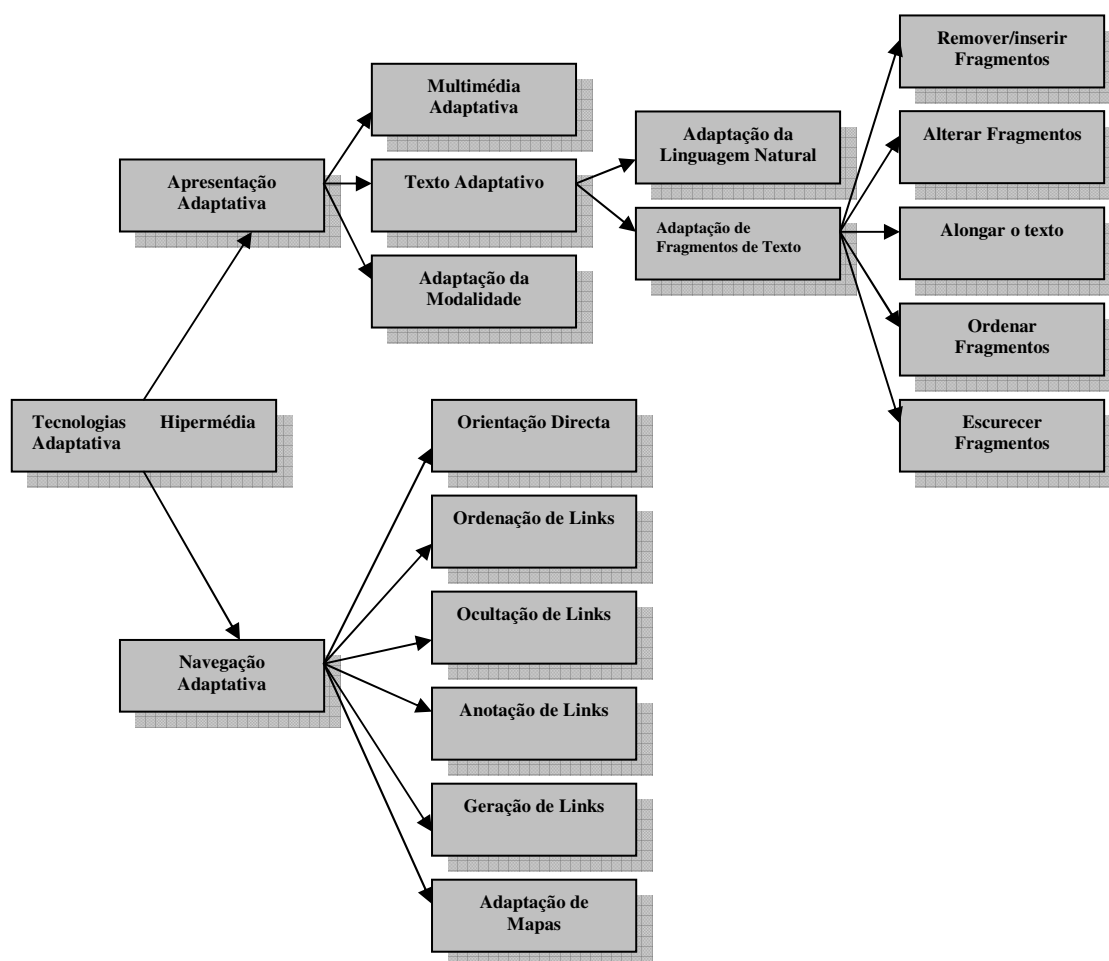


Figura 14 - Taxonomia das Tecnologias de Hipermedia Adaptativa, adaptado de [Brusilovsky, 01].

A Hipermedia Educativa refere-se aos sistemas que utilizam hipermedia adaptativa no ensino de um determinado domínio. O objectivo é fazer com que o processo ensino-aprendizagem atenda a todos os tipos de aluno, conforme as suas próprias características de aprendizagem. Os alunos podem navegar dentro do assunto do curso conforme as suas metas, conhecimento e preferências sem que seja necessário seguir um único caminho de aprendizagem previamente determinado pelo professor, um dos principais objectivos de sistemas tutores inteligentes.

A pesquisa em Hipermedia Adaptativa permite o desenvolvimento de mecanismos para facilitar e fornecer suporte ao desenvolvimento de Sistemas Tutores Inteligentes Adaptativos, por meio da definição de novas arquitecturas, ferramentas de Autoria e *Frameworks* Baseados em Componentes, razão da sua inclusão na *Framework* proposta neste trabalho, [Kobsa, 01].

2.16. Sistemas de Recomendação e Personalização

Um dos problemas que podem advir do uso de recursos computacionais está na sobrecarga de informação. Além disso, por vezes existem no contexto de uma organização, recursos que nem sempre são utilizados facilmente. Para muitos destes problemas a solução pode ser indicada por membros mais experientes, que nem sempre são identificados com facilidade. A mesma situação ocorre em sistemas de ensino, particularmente nos STI. Um sistema que procure exercer o papel de mediador entre os diferentes utilizadores, considerando as características individuais, e sugerindo recursos complementares para esclarecimento ou para o aprofundar de assuntos de interesse, procurando desta forma ampliar as zonas de desenvolvimento dos alunos e, portanto, potenciar novas aprendizagens, pode contribuir de uma forma significativa para a melhoria e eficácia de um STI. Trata-se no fundo de tentar replicar o que se passa na aprendizagem tradicional, onde os alunos trocam impressões entre si e o professor, dentro e fora da sala de aula, fornecendo sugestões de acordo com as suas experiências, recomendando recursos e assuntos a abordar, bem como a forma como o devem fazer.

Aprender é um acto cognitivo que envolve a aquisição de conhecimento e competências por parte de um aluno. Este acto é em grande parte uma experiência pessoal que pode ser realizada individualmente ou em grupo. No entanto, o sucesso dessa aprendizagem depende em grande parte das capacidades cognitivas do aluno, ou seja, da forma como a informação é armazenada e processada. Nesta área, os estudos científicos revelam a existência de diversos perfis cognitivos, o que leva a que a generalização do processo de aprendizagem se torne desadequada para a maioria dos alunos. É necessário assim, uma personalização do processo de aprendizagem, ou seja, adaptar o processo de aprendizagem a cada aluno. Num sistema de aprendizagem à distância a personalização pode ser efectuada de diversas formas, sendo que a infra-estrutura mais utilizada – a Internet – é um meio adequado para a recomendação e personalização.

Para tentar atingir estes objectivos, podemos recorrer a técnicas que são utilizadas nos chamados Sistemas de Recomendação (SR). A ideia básica de um SR é que sejam reconhecidas as preferências e/ou necessidades dos seus utilizadores sendo recomendado de forma automática aquilo que pode ser do seu interesse. SR têm vindo a ser utilizados em diversos contextos, preferencialmente no comércio electrónico, [Novais, 03].

O objectivo dos SR é o de que os utilizadores não apenas recebam o retorno de itens a partir da formulação de consultas, mas que o possível interesse do utilizador por um determinado item possa ser previsto. Esta tentativa de prever o que é mais adequado para um utilizador é uma forma de tentar evitar a sobrecarga de informações. Para implementação dos Sistemas de Recomendação existem várias abordagens:

- Filtragem por Conteúdos;
- Filtragem Colaborativa;
- Filtragem Demográfica;
- Abordagens Híbridas.

A Filtragem Baseada em Conteúdo parte do princípio de que os utilizadores tendem a interessar-se por itens similares aos que demonstraram interesse no passado. Assim é definida a similaridade entre os itens.

No caso da Filtragem por Conteúdos podemos ter:

- *Raw retrieval* - é um sistema que apenas apresenta um sistema de interrogações à base de dados, apenas apresentando a informação que o utilizador pretende;
- *Manually selected* - não utiliza qualquer tipo de computação apenas reproduzindo as preferências já pré-estabelecidas;
- *Statistical summaries* - contém medidas de popularidade e número de pessoas que recomendaram um determinado item. Exemplo: O item “X” foi utilizado por 10 pessoas, tendo 50% bom e 30% muito bom e 20% mau.
- *Attribute-based* - utiliza os atributos de um determinado item já manipulado para sugerir outras possibilidades. Exemplo: O utilizador X consultou itens sobre Ficção Científica então o sistema vai-lhe recomendar todos os recursos que tenham como género Ficção Científica.
- *Item-to-item correlation* - este tipo de técnica é utilizada para encontrar produtos que contenham associações entre eles, ou que sejam

minimamente similares, ou que sejam complementares. Exemplo: O utilizador X demonstrou interesse por bicicletas, então o sistema ao verificar todos os itens que contenham similaridades com a bicicleta, sugere ao utilizador colete reflector, capacete, luvas, travões de disco, etc.

O termo Filtragem Colaborativa está originalmente ligado a um sistema denominado Tapestry, [Goldberg et al, 92].

Esta abordagem procura verificar utilizadores que têm gostos semelhantes. Assim, no caso de um sistema que recomende filmes, a recomendação de um filme será feita a um utilizador se esse filme tiver sido bem avaliado por algum utilizador com perfil semelhante ao seu. Utilizadores com perfil semelhante, neste caso, são utilizadores que no passado avaliaram os mesmos filmes de maneira similar. As técnicas de Filtragem Colaborativa levam em conta que no dia-a-dia, é comum as pessoas recomendarem ou pedirem recomendações quanto a filmes, CDs, livros, software bem como nas actividades ligadas ao ensino. Na Filtragem Colaborativa, o principal problema reside no facto de que os novos itens que surgem somente são recomendados após terem sido avaliados por algum utilizador. Deve ser considerado que os sistemas de recomendação que utilizam a Filtragem Colaborativa apresentarão melhores resultados à medida que um número considerável de utilizadores avaliar os mesmos itens. A aplicação desta abordagem pode portanto ser um problema para os casos em que existem utilizadores com preferências bastante distintas, [Balabanovic, Shoham, 97].

No caso da Filtragem Colaborativa podemos ter:

- *User-to-user correlation* - o utilizador contém uma série de gostos/preferências que o inserem dentro de um grupo, pode-se então utilizar técnicas de filtragem para determinar quais os itens que devem ser recomendados ao utilizador, tendo em conta o historial de visualização dos outros utilizadores que se inserem no mesmo grupo. Exemplo: O utilizador X gosta de Matemática e os utilizadores Y e Z também gostam de Matemática, e tanto o utilizador Y como o utilizador Z utilizaram o item A, logo o que o sistema vai recomendar ao utilizador X é o item A.

No caso da filtragem demográfica, esta abordagem usa descrições de indivíduos (tais como: idade, ocupação, sexo) para determinar relações entre um determinado item e o tipo de indivíduo a que pertencem.

- Perfis criados a partir da classificação de utilizadores usando descrições convencionais e generalistas (estereótipos). Itens semelhantes são sugeridos para indivíduos com perfis demográficos semelhantes. Se as preferências ou interesses dos utilizadores mudaram constantemente ou rapidamente, os perfis demográficos não se adaptam a estes contextos. Os filtros demográficos raramente são utilizados independentes de outras quaisquer técnicas de filtragem.

As abordagens anteriores tendem a evidenciar fraquezas e tendem a ser complementares uma às outras. Na realidade não existem sistemas puros, mas sim combinações dos mesmos, daí as abordagens híbridas.

SR podem pois ser utilizados para facilitar o processo de ensino aprendizagem, através por exemplo de um assistente virtual que além de interagir com os utilizadores usando linguagem natural, procura determinar conteúdo a ser recomendado aos utilizadores a partir do acompanhamento das suas acções, bem como da comparação entre eles. Assim, o sistema constata que para resolver determinado exercício, disponibilizado no ambiente da ferramenta, a maioria dos alunos acedeu a determinados conteúdos por exemplo, como referido em, [Reategui, Lorenzatti, 05].

Existe um aspecto bastante importante, além da descrição dos métodos que estão por detrás da constituição de um SR, ou seja, o tipo de personalização que o SR pode fornecer. Uma das ideias subjacentes a este tipo de sistemas é a possibilidade de se adequarem em função de cada consumidor. Basicamente podemos falar de três tipos de personalização, como referido em, [Schafer et al, 01]:

- *Não personalizada* - em que não existe qualquer tipo de personalização ou adequação de resultados ao utilizador;
- *Personalização comum* - é um nível acima da não –personalizada devido ao sistema fazer recomendações que têm em conta as escolhas presentes do utilizador;

- *Personalização persistente* - este sistema cria recomendações que tenham em conta quer as escolhas que o utilizador está a fazer, neste momento, quer o perfil do utilizador. Neste nível de personalização, o tipo de técnica utilizada é a “user-to-user correlation”.

A personalização é o próximo passo na evolução dos sistemas de e-learning e dos STI em particular. Alguns investigadores propõem sistemas que proporcionam um certo grau de personalização aos utilizadores, por exemplo através do uso de ontologias, ou através da modelação inteligente de utilizadores, sendo propostas duas arquitecturas de personalização, Figura 15 e Figura 16, como referido em, [Gomes et all, 05], [Santos et all, 06].

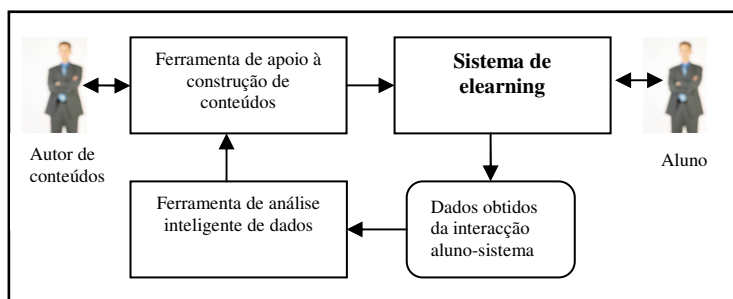


Figura 15 - Arquitectura para personalização off-line, adaptado de [Reategui, Lorenzatti, 05].

A personalização off-line incide sobre a construção dos conteúdos através da análise dos dados recolhidos da interação entre o aluno e o sistema, para depois aconselhar a equipa de desenvolvimento dos conteúdos sobre qual a melhor forma de construir estes mesmos conteúdos.

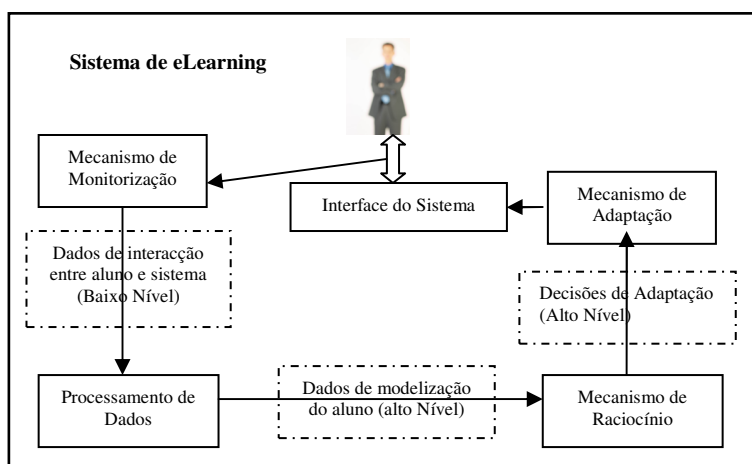


Figura 16 - Arquitectura para personalização on-line, adaptado de [Santos et all, 06].

A vertente on-line da personalização do sistema baseia-se na adaptação da navegação e dos conteúdos apresentados ao aluno durante a utilização do sistema. Para isso o sistema monitoriza constantemente o comportamento do aluno de forma a tentar identificar o seu estado, decidindo em cada caso qual a melhor opção a tomar quanto à adaptação do sistema ao aluno.

Um sistema de recomendação e personalização envolve explorar novos avanços nas tecnologias de computação, e a entrada da computação em novas áreas como sendo a assistência personalizada no ensino. No entanto, num futuro próximo, é impossível antecipar quais as necessidades de cada indivíduo a todo o tempo e em qualquer lugar, os sistemas de recomendação e personalização vão-se alastrar por todo o tipo de sistemas de computação que possam existir e ajudar a tomar decisões quando existirem demasiadas opções a verificar pelo utilizador, obviamente também nos STI.

2.17. Raciocínio Baseado em Casos

A possibilidade de um sistema aprender a partir da observação e da experiência é uma funcionalidade básica de qualquer sistema inteligente; ou seja, quaisquer entidades que se tornam conscientes do ambiente que as cerca, que desenvolvem crenças sobre os objectos e os acontecimentos no seu espaço de percepção, dão corpo a um sistema inteligente.

Ora, esta faculdade de acumular conhecimento como ajuda à resolução de problemas, apresenta-se como das facetas mais marcantes da actividade humana. Facultar aos sistemas computacionais este *talento* é um dos objectivos desta área do conhecimento.

O estudo do Raciocínio Baseado em Casos (RBC), surge com duas motivações: uma motivação cognitiva, que consiste em desenvolver sistemas que emulem o comportamento humano, e uma motivação pragmática, que consiste em desenvolver sistemas inteligentes que se mostrem eficientes, [Schank, 82].

Por outro lado, o senso comum leva-nos a concluir que o homem utiliza experiências passadas na resolução de problemas do presente, que a experiência melhora o seu desempenho, e que o conhecimento que tem sobre um assunto é por definição incompleto. Não é por conseguinte de admirar que o RBC se assuma como um dos paradigmas analíticos, para a resolução de problemas, mais promissores no campo da Aprendizagem Automática, daí a sua importância para os STI.

O advento do RBC fica a dever-se em grande parte a trabalhos de Roger Schank e Janet Kolodner. A partir de então o desenvolvimento de modelos e métodos para o RBC tem sido constante, [Schank, 82], [Kolodner, 88].

O processo de Raciocínio Baseado em Casos consiste, dado um problema, em encontrar e justificar a solução encontrada, o que passa pela reconstituição de uma experiência ou situação passada, reutilizando e adaptando o conhecimento de tal experiência ou situação a um novo caso, passível de vir a ser inserido na memória de casos.

Como exemplos de RBC podemos ter:

- Um médico, quando examina um doente, aconselha uma terapêutica para o tratamento da enfermidade detectada e, então, regista esse conhecimento sobre a forma de casos. Entretanto, novos pacientes afluem ao seu consultório. Assumindo que existem analogias entre as patologias indiciadas pelos casos da memória de casos, e o novo caso, o do paciente em observação, o clínico usa o conhecimento do passado para tratar o doente do presente;
- Um mecânico com experiência, com base na observação e no conhecimento do tipo de ruídos que um veículo pode apresentar, determina qual o problema e também qual a reparação a efectuar;
- Um professor perante um aluno com dificuldades, usa a sua experiência de ensino (de casos passados similares), para utilizar métodos e estratégias que demonstraram validade no passado, evitando aquelas que não se revelaram úteis.

O raciocínio por reutilização de casos passados é, sem dúvida, uma poderosa ferramenta na resolução de problemas. No RBC, um caso indica uma situação, uma experiência passada cujo conhecimento foi capturado e assimilado de tal forma que pode ser usado na resolução de problemas futuros.

Uma das características mais importante do RBC é a sua ligação à aprendizagem. No RBC opta-se não só por um método de raciocínio, mas também por um processo de aprendizagem suportado por uma actualização de uma base de conhecimento, após concluída a resolução de um problema. A aprendizagem ocorre naturalmente através da resolução de casos. Quando um problema é resolvido, a experiência é retida de forma a que se resolvam casos similares no

futuro. Quando a resolução falha, a razão da falha é identificada e lembrada de modo a evitar-se, futuramente, a repetição dos mesmos erros.

O raciocínio baseado em casos relaciona-se com a área dos STI por duas vias:

- O RBC constitui um modelo cognitivo e de aprendizagem, o que sugere a ideia de que o objectivo de um tutor inteligente deveria incluir ensinar a partir de casos, bem como ser capaz de os indexar convenientemente. As regras são úteis ou mesmo essenciais, mas parecem apenas tornar-se intuitivas a partir do momento em que existe uma base rica de exemplos mostrando como as regras se aplicam e em que circunstâncias devem ser aplicadas;
- Por outro lado, o RBC pode ser considerado como uma tecnologia para construir STI. Esta tecnologia enfatiza a construção, a indexação e o uso de grandes bibliotecas de exemplos relacionados, ainda que eventualmente redundantes, em contraponto ao desenvolvimento de motores de inferência e elaboração de conjuntos de regras.

Os processos de RBC têm como objectivo, a partir de uma análise prévia de um problema, encontrar casos similares, usar casos para sugerir uma solução para o problema, avaliar a solução proposta e actualizar o sistema (Base de Conhecimento), aprendendo com esta experiência.

Existe um conjunto de diferentes métodos para organizar, recuperar, e utilizar o conhecimento passado retido em casos. Os casos podem ser guardados como experiências isoladas ou agrupados como conjuntos de casos similares na forma de um caso geral. Os casos podem ser armazenados separadamente ou divididos em sub-unidades e apresentarem-se estruturados. A solução de um caso anterior pode ser aplicada directamente ao problema em mãos, ou modificado de acordo com as diferenças entre os dois casos. A adaptação de soluções e formas de aprendizagem com base em casos pode ser suportada e guiada por formas de representação de conhecimento e raciocínio do domínio geral ou, muito simplesmente, baseado em aparentes similaridades sintácticas entre os casos. Um RBC pode ser automático ou interactivo, estando, nesta última variante, o utilizador com o controlo das operações.

As memórias de eventos passados são usadas directamente nas tomadas de decisão. Raciocinar fica assim reduzido à percepção, ou seja, a situação corrente é observada, procura-se na memória de casos os que mais se aproximam da situação corrente e, utiliza-se esse

conhecimento para a resolução do problema. Os casos são organizados vectorialmente, denotando atributos e relações, de modo análogo ao que acontece na criação e manutenção das bases de dados. As formas de organização e acesso à memória de casos são peças centrais para a análise de desempenho deste tipo de sistemas.

Resolver um problema, que é passado como um caso, não passa além de um simples processo de classificação, ou seja, há que procurar a classe que irá reter o caso não classificado. Nessa classe ir-se-á procurar, então, uma solução para o problema. O conjunto de classes torna-se, nesta situação, no conjunto das possíveis soluções. As tarefas de classificação são aqui efectuadas a partir de casos passados, em vez de o serem através de regras, como é habitual nos sistemas tradicionais. Um sistema como o *Protos*, destinado ao diagnóstico de patologias auditivas, é um exemplo de aplicação deste método.

Este tipo de raciocínio é particularmente adaptado a domínios cuja formalização em regras seja difícil ou mesmo impraticável. Assim, nestes casos raciocinar com base em exemplos parece ser a única alternativa válida. Outro campo privilegiado de aplicação é o de domínios em que, pelo contrário, existe um número de regras demasiado elevado, como seja economia ou meteorologia, ou em que as regras existentes podem ser aplicadas de demasiadas formas diferentes, como a matemática ou a programação. Em tais domínios, os casos ou exemplos sugerem respostas aproximadas, limitando assim o número de combinações de regras a explorar.

Este modelo de raciocínio baseia-se em três processos fundamentais:

- *Indexação* ou etiquetagem – quando uma nova experiência é fornecida ao sistema, ela deve ser classificada ou "etiquetada" para poder ser futuramente recuperada;
- *Recuperação* – selecção da experiência pretendida, de entre as várias possíveis que se adaptam à situação;
- *Adaptação* – quando esse conhecimento é seleccionado com vista a resolver um dado problema, deverá ser adaptado de modo a poder ser aplicado à nova situação.

Há necessidade de indexar os casos de duas maneiras:

- A indexação *concreta* refere-se aos objectos e acções que normalmente são mencionados de forma directa no caso;

- A indexação *abstracta* refere-se a uma caracterização mais geral do caso. O problema da indexação, ou seja, de como determinar os índices concretos e abstractos apropriados para um dado caso, está longe de ser trivial. A forma como essa indexação é feita determina quais os casos que vão ser comparados com uma dada nova situação em estudo. Por exemplo, uma indexação muito generalista vai fazer com que sejam fornecidos casos que não partilham detalhes específicos com a situação em análise.

A Figura 19 detalha os componentes de raciocínio e de aprendizagem de um sistema baseado em casos típico, [Riesbeck, Schank, 91]:

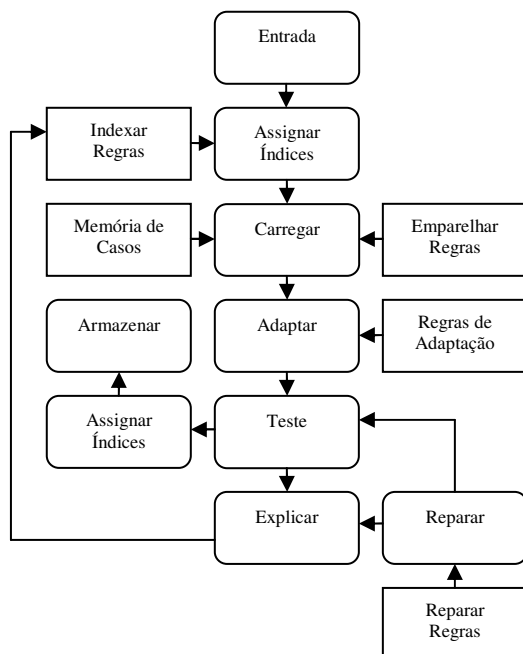


Figura 17 - Modelo ilustrativo do raciocínio baseado em casos, baseado em [Riesbeck, Schank, 91]

O processo utilizado é basicamente o seguinte:

- A informação descrevendo um problema ou situação é analisada e usada para seleccionar um ou mais casos similares na biblioteca de casos;
- A solução utilizada no caso mais parecido é de seguida adaptada para resolver o problema em estudo;
- Se tal solução se mostra válida, é então acrescentada (juntamente com o caso em apreço) à biblioteca de casos;

- Se a solução proposta falha, podem ocorrer duas situações:
- A falha é analisada e a solução corrigida, após o que é de novo testada e, caso funcione, é acrescentada à biblioteca de casos;
- Novos índices são criados, baseados na análise de falhas, de modo a que, de futuro, seja utilizada a solução corrigida e não a original.

Idealmente, um sistema baseado em raciocínio por casos não deveria nunca repetir o mesmo processo de resolução de problemas nem, por outro lado, deveria repetir os erros cometidos.

Não parece ser aconselhável efectuar a indexação por palavras ou frases, sendo as razões mais importantes as seguintes:

- Teriam que ser incluídos sinónimos para cada palavra contida no índice;
- Os usos comuns de certas palavras podem não coincidir com os significados presentes na biblioteca de casos, ou seja, palavras percebidas como sinónimos pelo uso comum podem não o ser na definição mais rigorosa contida nessa biblioteca;
- Palavras ambíguas levariam a seleccionar casos sem relação efectiva;
- Os casos deveriam ser reindexados sempre que a linguagem comum da população estudantil mudasse;

Há ainda que definir como é que um dado conceito presente no índice se relaciona com um dado caso. Poder-se-á, por exemplo, utilizar relações do tipo "uma instância de ...", "a causa de ..." ou "é uma excepção a ...", em que as reticências são substituídas por conceitos.

2.18. Teorias de Aprendizagem

A ideia de utilizar a tecnologia, e particularmente a computação no campo do ensino, tanto nas escolas como nas universidades não é nova. A instrução assistida por computador (IAC) integrou-se perfeitamente, devido ao alento fornecido pelo baixar dos custos de hardware e de acesso à Internet, [Yang et al, 02]

Durante o processo de aprendizagem, pretende-se promover a interacção entre o aluno e o professor, (neste caso personificado pelo Sistema Tutor Inteligente), pelo que seria adequado

um sistema que permitisse “ múltiplas estratégias” (à semelhança do tutor humano). Isto traria robustez e o nível de adaptação desejado às características mutantes dos estudantes.

Castillo estabelece que a pedagogia influencia o desenvolvimento dos sistemas de ensino, que devem basear-se no modelo do aluno, assim como as estratégias de ensino efectivo nas teorias de aprendizagem, abarcando portanto o modelo do aluno e o modelo do tutor (estratégias de ensino). Refere que “*a psicologia intervém no estabelecimento de modelos cognitivos, necessários para o entendimento e aplicação apropriada dos modelos de ensino citado*”, para dar conta que para ensinar se deve conhecer como os sujeitos conhecem, [Castillo et al, 97].

A necessidade de que a instrução assistida por computador se transforme num apoio efectivo para a tutoria, requer uma base teórica que sustente o desenvolvimento dos sistemas, para que estes sejam efectivos no processo de ensino-aprendizagem. Existem diferentes teorias que explicam o processo de aprendizagem, que se podem agrupar em:

Instrucional: onde o professor ou tutor é o centro da aprendizagem e responsável por desenhar o conteúdo, relegando o aluno para a função de “aprender” o conhecimento que o professor lhe comunica. Este processo de aprendizagem é altamente guiado pelo professor e não requer iniciativa pessoal do aluno.

Aprendizagem por descoberta: onde o professor tem uma tarefa de “explorador”, entregando ao aluno as ferramentas necessárias para que este descubra o conhecimento, ao seu próprio ritmo. Este processo de aprendizagem não é guiado, e requer iniciativa por parte do aluno, já que é ele que será responsável por “descobrir” o conhecimento pelos seus próprios meios, guiado pelo tutor para que se possam atingir os objectivos da sessão pedagógica planeada, mas dando lugar a novos objectivos que surgem através da experimentação do aluno.

As primeiras aplicações de IAC ocorreram desde a visão comportamentalista. Skinner, muito influenciado pelas investigações de Pavlov⁶, define o comportamentalismo não como uma forma de estudar a conduta, senão como uma filosofia da ciência dedicada ao objecto e métodos da psicologia. Skinner refere que o comportamento dos organismos pode ser explicado através das contingências ambientais, e os processos internos de natureza mental não tem nenhum poder

⁶ Pavlov, médico russo, quando alimentava os seus cães de laboratório introduzia o ruído de uma campainha. Por meio de medições comprovou que o som da campainha provocava salivação nos cães, mesmo que o som não viesse acompanhado de comida. Depois de anos avaliando os resultados experimentais, estes dariam lugar à teoria conhecida como condicionamento clássico de Pavlov, que seria mais tarde a base do comportamentalismo, escola psicológica que pretende explicar e prever o comportamento, que trás alguma luz sobre muitas das aprendizagens inclusivamente em humanos.

causal-explicativo. Na actualidade o comportamentalismo evolui para modelos cognitivos mais flexíveis e adequados que põem em primeiro plano o aluno no processo educativo.

De seguida resumem-se os modelos mais representativos e referem-se as suas características principais, dada a sua importância e ligação inerente ao desenvolvimento de STI.

2.18.1. Teoria da Instrução de Gagné

Nesta teoria, a aprendizagem define-se como um processo que permite ao organismo vivo modificar os seus comportamentos de forma rápida e permanente; portanto, a aprendizagem verifica-se quando existe uma mudança de comportamento, relativamente estável. O que supõe quatro elementos:

- Um aprendiz
- Uma situação que permita a aprendizagem
- Um comportamento explícito do aprendiz
- Uma mudança interna

Esta teoria estabelece que existem vários tipos de aprendizagens, o que torna necessários diferentes modos de instrução. Esses tipos são:

- Informação verbal
- Destrezas intelectuais
- Discriminar, formar conceitos
- Aprender regras
- Estratégias cognitivas
- Atitudes e destrezas motoras

Para cada tipo de aprendizagem são necessárias diferentes condições internas e externas. Uma vez que apresenta estes factores que afectam a aprendizagem, Gagné pode apresentar o processo de aprendizagem como um sistema, onde as entradas afectam as saídas, e portanto o

professor pode melhorar a saída controlando as entradas. Gagné sugere agregar o conceito de hierarquia como base para a sua descrição de funcionamento do sistema.

2.18.2. Teoria do Equilíbrio de Piaget

Para Piaget, existem dois tipos de aprendizagem, sendo um a aprendizagem em sentido estrito, através da qual se consegue informação específica e outro, a aprendizagem em sentido amplo, que consiste no progresso das estruturas cognitivas. A aprendizagem produz-se quando se apresenta um desequilíbrio ou conflito cognitivo, que dá lugar a dois processos: a assimilação e a acomodação, [Piaget, 89].

A assimilação é a integração de elementos exteriores nas estruturas em evolução ou já acabadas no organismo. O indivíduo interpreta a informação do meio de acordo com os seus conceitos disponíveis. Através do processo de acomodação, o sujeito adapta os seus conhecimentos à realidade contrastando com a mesma realidade o que assimilou; por outra parte, a mudança de estruturas cognitivas do indivíduo é produzida em dois sentidos, o primeiro como consequência da soma de novos conceitos e o segundo como consequência de uma reinterpretação do conhecimento existente à luz do novo conhecimento.

A teoria de Piaget reduz toda a aprendizagem a aquisições espontâneas e necessárias. Isto é uma falha dado que a maior parte dos conceitos na realidade não são necessários e ainda, não se podem adquirir sem a intervenção da cultura e da instrução, ou seja não são espontâneos.

2.18.3. Teoria Sociocultural de Vigotsky

Vigotsky refere que “ estudar a conduta do homem sem a parte psíquica, como pretende a psicologia, é tão impossível como estudar o psíquico sem a conduta. O que implica que não existe lugar para ciências distintas. O estado actual dos dois ramos do saber refere insistentemente a questão de tal necessidade e fecundidade da completa fusão de ambas a s ciências”, [Vygotsky, 78].

A aprendizagem sucede a partir da interacção social, pode estabelecer-se que: o individuo reconstrói o seu saber devido à mistura entre processos de construção pessoal e processos de colaboração com os outros intervenientes. Neste ultimo processo, o professor/tutor deve ser um

agente facilitador e mediador da cultura. Na Tabela 7 podemos analisar o paralelismo entre a concepção de Piaget e a de Vigotsky.

Tabela 7 – Teoria de Piaget vs Teoria de Vigotsky

Teoria de Piaget	Teoria de Vigotsky
O conhecimento é um processo de interação entre o sujeito e o meio (físico unicamente)	O conhecimento é um processo de interação entre o sujeito e o meio (social e cultural)
O ser humano ao nascer é um indivíduo biológico	O ser humano ao nascer é um ser social
No desenvolvimento do ser humano existe um processo de socialização	No desenvolvimento do ser humano existe um processo de diferenciação social
O potencial cognitivo do sujeito depende da etapa de desenvolvimento em que se encontra	O potencial cognitivo do sujeito depende da qualidade cultural da interação social e da ZDP do sujeito.
O ser humano ao nascer encontra-se num estado de desorganização que deverá sendo sucessivamente organizado ao longo das etapas de desenvolvimento da sua vida	O ser humano ao nascer tem uma percepção organizada pois está dotado para a dirigir para estímulos humanos e para estabelecer interações sociais

2.18.4. Teoria de Bruner

Esta teoria está relacionada com a teoria de Piaget, já que enfatiza a aprendizagem por descobrimento. O aluno descobre, pois a descoberta causa maior satisfação e obtém uma maior retenção. Chega um passo mais adiante que a teoria de Piaget, desenvolvendo um processo de instrução onde o professor deve criar um ambiente que favoreça o atingir da descoberta. Propõe que os conteúdos a ensinar se apresentem como um conjunto de problemas e relações que o aluno deve resolver para que aumente o seu interesse e resulte uma aprendizagem significativa, [Bruner, 90],[Guardia, 93].

Bruner explica como o aluno chega a dominar os conteúdos que o docente lhe propõe construindo o conceito de “andaime”, de “suporte”, estritamente ligado ao conceito de zona de desenvolvimento proximal ZDP de Vigotsky. Segundo esta ideia, no processo de interação e diálogo em que se baseia o ensino, o perito tem um conjunto de andaimes ou ajudas por meio das quais o aluno elabora as construções necessárias para aprender os conteúdos. O “andaime” deve contemplar as características particulares do aluno. Serão reguladas à medida em que as

suas habilidades vão progredindo. Assim, alguns iriam requerer apoio como explicações, modelos, etc, enquanto que outros necessitariam de apoios mais complexos, [Bruner, 90].

Progressivamente o professor/tutor ajustará o sistema de ajudas e apoios necessários conforme se desenvolvam as habilidades dos alunos, e assim poderá ceder o controlo e a manipulação dos conteúdos e formas de discurso a aprender.

2.18.5. Teoria da Aprendizagem Significativa de Ausubel

Esta teoria trata da aquisição de aprendizagens significativas, ou seja, não considera a aprendizagem mecanizada (de memória). Para considerar que uma aprendizagem é significativa, esta deve ter um sentido para ser incorporada no conjunto de conhecimentos que o sujeito possui (o conceito aprendido deve relacionar-se com conceitos previamente adquiridos). Considera também a aprendizagem como receptiva, já que o professor é quem estabelece os conteúdos de forma a facilitar a organização do novo conteúdo na estrutura mental do aluno.

Ausubel estabelece que para que se produza a reestruturação é necessária uma instrução que apresente a informação organizada e explícita que desequilibre as estruturas existentes. *Uma aprendizagem é significativa quando se pode incorporar nas estruturas de conhecimento, ou seja, os novos conceitos adquirem significado para o sujeito dentro das suas estruturas cognitivas.* Uma aprendizagem é “de memória” quando os conceitos aprendidos não estão relacionados entre si e carecem de significado para a estrutura do sujeito. Em geral uma aprendizagem significativa é mais eficaz que a aprendizagem de memória pois: produz retenção mais duradoura, facilita novas aprendizagens e produz mudanças que persistem para além do esquecimento dos detalhes concretos. Outra contribuição de Ausubel é o conceito de organizadores prévios, sobre os quais o aluno se apoia, de tal forma que estes fazem as vezes de estrutura ou andaime entre os conhecimentos a adquirir e os que já possuía.

2.18.6. Inteligências Múltiplas de Gardner

Esta teoria da inteligência humana, desenvolvida pelo psicólogo Howard Gardner estabelece que existem pelo menos oito formas através das quais as pessoas percebem e entendem o

mundo. Gardner chama a cada uma dessas formas *inteligências* (conjunto de capacidades que permitem resolver problemas da realidade), [Gardner, Hatch, 89].

Gardner define uma inteligência como “ a capacidade de resolver problemas ou de criar produtos que sejam valiosos num ou mais ambientes culturais” Este autor fundamenta a sua estrutura em provas biológicas e antropológicas, mais especificamente, em bases neurológicas, evolucionistas e trans-culturais. Não obstante, o autor clarifica que é uma aproximação que não estabelece as fontes de tais capacidades ou os meios para as medir. Gardner apresenta uma classificação de inteligências, ainda que sugerindo que não é uma classificação terminal.

- Verbal - linguística
- Lógico - matemática
- Visual - espacial
- Kinestésica
- Musical - rítmica
- Interpessoal
- Intrapessoal
- Naturalista

Para Gardner “ o objectivo da escola deve ser o de desenvolver as inteligências e ajudar as pessoas a atingir os fins vocacionais e as afinidades que se adequam ao seu espectro particular de inteligências.”

Este objectivo consegue-se com uma escola centrada nos indivíduos, onde a avaliação das capacidades e das tendências individuais é contínua.

Gardner assinala que a pratica educativa convencional se centra fundamentalmente na inteligência linguística e matemática, mas dado o carácter múltiplo da inteligência humana se deveria dar espaço às diversas capacidades e características das pessoas. Isto pode incluir-se num STI através de distintos protocolos pedagógicos que se adequem da melhor maneira a cada um dos alunos, [Gardner, Hatch, 89].

2.18.7. Teoria UNO de Perkins

Uma vez apresentadas as técnicas educativas mais importantes, deve-se analisar quais de entre as apresentadas são aplicáveis ao ensino e implementação num STI.

Qualquer das teorias de ensino expostas anteriormente resulta útil para entender o processo de aprendizagem num STI, a selecção de um marco teórico adequado só se pode justificar por meio de razões pedagógicas correspondentes ao tipo de alunos com os quais o sistema tutor inteligente se vai “enfrentar”, [Perkins, 95].

Perkins propõe a Teoria Uno que não é um modelo nem um método de ensino mas sim um conjunto de recomendações compatíveis com qualquer teoria. Basicamente estipula que “as pessoas aprendem mais quando têm uma oportunidade razoável e uma motivação para o fazer” e para aplicar esta teoria devem estar reunidas as seguintes condições: informação clara, prática reflexiva, realimentação informativa e uma forte motivação intrínseca e extrínseca.

Perkins refere que combinando as condições que estipula a “Teoria Uno” com cada um dos programas de estudo, obtêm-se os métodos respectivos. Por outras palavras, essa teoria “personifica-se” de distintas formas segundo o programa do momento. As quatro formas principais são enumeradas e brevemente explicadas de seguida.

- Instrução didáctica
- Treino
- Ensino socrático
- Outras

A instrução didáctica

Segundo Perkins, é a apresentação clara e correcta da informação por parte do professor e dos textos. O seu objectivo centra-se especialmente na explicação onde se expõe “os quês” e os “porquês” de um determinado tema.

É fundamental que se aclarem os componentes básicos de uma boa explicação, já que na instrução didáctica o aluno encontra-se renegado para um papel de sujeito passivo e o tutor (professor) fica em primeiro plano, como fornecedor único dos saberes e conhecimentos. Algumas características que deve reunir uma boa explicação na prática educativa são: a

identificação de objectivos para os alunos, a supervisão e o assinalar do caminho até aos objectivos, mostrar vários exemplos sobre os conceitos analisados, legitimar um novo conceito ou procedimento mediante princípios já conhecidos pelos alunos.

As aulas do tipo prático não ficam fora deste modelo, apenas nelas se tentam incluir exposições complementares, para assinalar vínculos entre os conceitos familiares e os novos conceitos, clarificando as condições de aplicabilidade e de não aplicabilidade dos mesmos. A interacção é sempre liderada pelo professor ou tutor, renegando o aluno para um segundo plano onde o seu único papel é o de assimilar a informação que lhe está a ser apresentada no momento, portanto os elementos básicos da instrução didáctica têm sobretudo a ver com a clarificação e clareza informativa

O treino

Existe um vínculo entre o treino e a instrução didáctica e por isso este é apresentado em segundo lugar. Sem uma instrução didáctica que apresente uma certa base de informação clara sobre os novos conceitos, os estudantes não iriam possuir os conhecimentos básicos para realizar exercícios de cariz prático. Surge então uma aparente contradição que é, dada a clareza informativa da instrução didáctica, que função passa a desempenhar o professor (tutor), se numa forma ideal, a instrução didáctica resolveu todas as dúvidas possíveis sobre o conceito explicado?

O treino oferece uma resposta, já que se baseia em duas das condições apresentadas por Perkins na sua teoria: a prática reflexiva e a re-alimentação informativa.

As principais actividades do docente que proporciona o treino consistem em:

- Atribuir tarefas práticas;
- Impulsionar os alunos a reflectir sobre as tarefas que estão a realizar;
- Fornecer a retroalimentação.

Ao mesmo tempo, espera-se que o docente responsável pelo treino forneça a informação de forma clara dado que é a relação entre o “treinador” e os seus alunos a que deve fomentar mecanismos de motivação destes últimos e garantir assim uma das questões fundamentais da teoria de Perkins que afirma: “as pessoas aprendem mais quando têm uma oportunidade razoável e uma motivação para o fazer”.

O ensino socrático

Ambos os métodos anteriormente vistos (a instrução didáctica e o treino) possuem um aspecto regulativo, pois a sua função consiste em modelar e guiar as actividades dos alunos e como se explicou anteriormente renega-se o aluno para um papel secundário, tendo como única função a de assimilar os conhecimentos que se apresentam pelo docente ou tutor.

Neste marco surge o método socrático através do qual se tenta que os alunos trabalhem de uma maneira mais flexível, mas não de forma totalmente livre, continuamente deverão receber o apoio nas suas investigações por parte do tutor, mas sem o esquema mais rígido em que este lhes diz continuamente o que têm de fazer. Perkins apresenta que, desta forma o estudante deve aprender sozinho as respostas, mas também “ a arte das perguntas”.

O ensino realiza-se quando o “mestre socrático” apresenta um enigma conceptual e incita a investigação do assunto de forma livre por parte dos alunos, fazendo perguntas do tipo:

- Que pensam a esse respeito?
- Que posição se poderia tomar?
- De que definições necessitamos?

Logo na investigação inicial propõem-se ideias, critérios e definições, onde o tutor actua como incitador e moderador do diálogo/discussão: presta ajuda quando os paradoxos estancam o processo de aprendizagem e gera contra exemplos e potenciais contradições quando percebe nos alunos uma satisfação prematura. Desta forma a aprendizagem é “guiada” pelos tutores até se atingir um objectivo, já que sem um guia eficaz os estudantes podem perder-se. Isto não se faz de forma rígida, como na instrução didáctica e no treino, já que no método socrático se espera que seja o aluno que realize as descobertas por si mesmo. O investigador cognitivo Collins analisou os passos fundamentais do método socrático que se descrevem de seguida, [Perkins, 95]:

- Seleccionam-se exemplos positivos e negativos para ilustrar as qualidades pertinentes ao tema que se esta a abordar;
- Variam-se os casos sistematicamente afim de centrar a atenção em dados específicos;
- Empregam-se contra exemplos para por ajuizar das conclusões do aluno;

- Propõem-se casos hipotéticos para que o aluno reflecta sobre situações afins que poderiam ocorrer naturalmente;
- Utilizam-se estratégias de identificação de hipóteses para forçar a articulação de uma hipótese específica de trabalho;
- Empregam-se estratégias de avaliação de hipóteses para fomentar a avaliação crítica de previsões e hipóteses;
- Promove-se a identificação de outras previsões que poderiam explicar o fenómeno em questão;
- Utilizam-se estratégias facciosas para induzir o aluno a realizar previsões incorrectas e formulações prematuras;
- Procura-se que o aluno deduza as consequências até chegar a uma contradição para que aprenda a construir teorias válidas e consistentes;
- Questionam-se as respostas provenientes de referências como o tutor e os manuais, afim de promover o pensamento independente.

Segundo Perkins, “ o mestre socrático não deve fornecer abundância de dados, mas sim controlar a qualidade e a clareza da informação fornecida pelos alunos fazendo-lhes perguntas. Quando os alunos discutem entre si sobre uma determinada questão, o mestre socrático exige-lhes uma prática continua de reflexão. Fornece ainda re-alimentação imediata por meio de estímulos e crítica, e incita a que todos os participantes da conversa façam o mesmo.

Outros métodos de ensino

Os protocolos pedagógicos propostos por Perkins não são as únicas aproximações que provêm das características referidas pela sua teoria: *“informação clara, pratica reflexiva, re-alimentação informativa e forte motivação intrínseca e extrínseca”*.

Existem outras aproximações como podem ser a aprendizagem cooperativa e a colaboração entre pares, que vale a pena referir no campo dos assessores inteligentes, onde o sistema deve reagir como um “estudante virtual” que interactua com os alunos que utilizam o sistema para resolver os distintos problemas que se lhe apresentam e para atingir definições básicas para os conceitos que se pretendem ensinar numa sessão pedagógica.

Perkins refere que as crianças aprendem muito melhor em grupos cooperativos bem configurados que sozinhos. De uma forma geral, os grupos cooperativos podem ajudar a atingir determinados fins, sem a planificação cuidadosa que requerem os métodos como a instrução didáctica e o treino. Os resultados obtidos são satisfatórios porque a aprendizagem cooperativa exige que todos os participantes sejam responsáveis pelo desempenho do grupo.

A aprendizagem cooperativa e a colaboração entre pares podem utilizar a dinâmica de grupos para promover a aprendizagem reflexiva, onde os alunos pensam e discutem os conceitos que se lhes tentam ensinar e os problemas que devem resolver. Na aprendizagem cooperativa e colaborativa, a predisposição para a resolução de tarefas, eleva a motivação do contacto social para manter os alunos interessados nas suas actividades académicas.

Pode parecer que na aprendizagem cooperativa e colaborativa o docente não possui um plano que deva cumprir, e portanto este método de ensino não seria aplicável aos Sistemas Tutores Inteligentes mas, uma análise mais profunda revela que os docentes têm as seguintes tarefas, [Perkins, 95]:

- O docente “treina” os alunos para que julguem com critérios próprios adequados à circunstância planeada;
- O docente deve “aumentar” o interesse intrínseco dos problemas que o mesmo apresenta;
- O docente pode “pedir aos alunos que se classifiquem mutuamente”, mas oferecendo uma forte re-alimentação relativamente a que tipo de respostas tem mais ou menos sentido e porquê.

2.19. Base de Conhecimento

Na *Framework* apresentada, a Base de Conhecimento (BC), foi colocada “fora” do STI de forma a permitir que o sistema seja independente do seu domínio de actuação, ou seja, procura-se que os mecanismos utilizados para o ensino sejam independentes da matéria objecto de ensino. A existência de uma BC externa ao STI, permite a necessária adaptabilidade a qualquer domínio de ensino, assumindo que o conhecimento se encontra representado e armazenado de forma a permitir a sua manipulação pelo STI, independentemente do seu conteúdo específico. Esta característica torna-se bastante ambiciosa, provavelmente demais até, ao sugerir de certa

forma que um STI pode ser completamente independente do domínio em que se insere, seria o mesmo que pensar que um professor deveria ser capaz de ensinar qualquer domínio independentemente da sua formação de base.

2.20. Base de Dados de Modelos Pedagógicos

No processo de ensino-aprendizagem utilizam-se diferentes métodos de ensino, baseados em diferentes teorias de aprendizagem. Na procura de uma maior flexibilidade e efectividade no ensino, a *Framework* FD-STI incorpora uma base de dados de modelos pedagógicos passíveis de serem utilizados. Pretende-se que, através da caracterização que o sistema faz do aluno, (através do modelo do aluno, cap. 2.6), os métodos de ensino mais adequados sejam utilizados de acordo com o modelo do aluno em questão. Adicionalmente, para um mesmo aluno, e caso um método de ensino se revele menos adequado, o sistema deverá possuir a capacidade de trocar de método. O facto de se encontrarem numa base de dados, num formato que permite a manipulação pelo sistema, permite também que sejam incorporados novos modelos, sem que o sistema tenha de ser alterado, como aliás se processa também ao nível da base de conhecimento.

Ao colocar uma base de modelos pedagógicos fora do próprio STI, está-se também, e mais uma vez, a reiterar a importância das teorias da aprendizagem, da pedagogia, num tal sistema, tornando-o dinâmico na forma como ensina, ou seja, o professor ideal que muda de estratégias e forma de actuação perante situações e alunos diferentes.

2.21. Conclusões

Ao longo deste capítulo, foram enumeradas as características desejáveis de um STI, bem como aqueles que se julga serem os problemas mais pertinentes inerentes ao seu desenvolvimento e aplicação. Do trabalho desenvolvido, ressaltam algumas questões que constituem parte dos problemas que carecem de resolução no desenvolvimento de STI, e que foram sintetizados numa matriz problemas/contribuições. Como forma de contribuir para a possível resolução das questões enunciadas, a FD-STI visa pois propor técnicas e métodos que se julgam importantes para o desenvolvimento de STI, contribuindo com melhorias fundamentais. Após detalhar e aprofundar os componentes da FD-STI, as possíveis contribuições de cada um deles foram explicitadas e referenciadas.

Também importante, como referido ao longo deste trabalho, é o envolvimento de outros ramos da ciência que não exclusivamente a informática, como as ciências da educação e a psicologia. As teorias de ensino devem ter um papel preponderante nos STI, pois permitem ao sistema socorrer-se de um teoria de ensino e adaptar-se ao aluno e à sua forma de aprender, deixando de ser o aluno responsável por se adaptar ao sistema, mas sim o sistema que se deve adaptar. Esta mudança de atitude e de perspectiva contribuirá para melhorar este processo de ensino.

Conclusões

Esta dissertação reflecte um trabalho de levantamento sobre a área de Sistemas Tutores Inteligentes, desde o seu aparecimento como área de investigação, das suas etapas de evolução até ao que se designa hoje como sendo um Sistema Tutor Inteligente. Os conceitos relacionados foram introduzidos, e as arquitecturas relevantes foram estudadas. Partindo da arquitectura denominada clássica ou tripartida, os módulos que a constituem foram enumerados e dissecados, procurando focar e contextualizar os aspectos mais importantes. Da análise dos Sistemas Tutores Inteligentes desenvolvidos mais relevantes, ressaltaram alguns pontos fortes mas também algumas debilidades permitindo identificar as características que se julgam desejáveis num tal sistema, bem como alguns erros a evitar. Sugeriu-se a necessidade de investigação em diferentes domínios que não apenas a Informática, dada a natureza multidisciplinar dos STI, desde logo a Psicologia e as Ciências da Educação, dado que se pretende intervir no processo de ensino. Pode-se verificar que, os STI foram condicionados pelas correntes pedagógicas da altura em que foram desenvolvidos, procurando pois reproduzir o tipo de ensino em vigor à data.

A Inteligência Artificial assume um papel preponderante para tornar possível a adaptabilidade dos STI. Os STI surgem como potenciadores do processo de ensino e não como substitutos do tutor humano, ficando exposto que a utilização de STI produz resultados interessantes relativamente a grupos de alunos que não os utilizam, mas ainda assim inferiores aos resultados obtidos por grupos de alunos com um tutor individualizado.

Vários Sistemas Tutores Inteligentes foram analisados, ressaltando que o caminho a percorrer ainda é longo, pois o nível de inteligência de tais sistemas ainda se encontra aquém do esperado. Verifica-se na grande maioria dos STI uma dependência do próprio sistema com o seu domínio de actuação, um dos factores que limita e muito a escalabilidade de tais sistemas. Exemplo disso é o sistema CIRCSIM, um sistema efectivamente utilizado e com níveis interessantes de sucesso. No entanto, encontra-se fortemente dependente do seu domínio de actuação, não sendo adaptável a outros domínios. Uma possível excepção é o AUTOTUTOR, com a sua aproximação por “Curriculum Scripts” e “Authoring Tools” que pretende especificamente a independência do sistema relativamente ao domínio em estudo. Este é o sistema que mais perto se encontra dos objectivos desejados para um STI.

Enumeradas as características desejáveis de um STI, bem como aqueles que se julga serem

os problemas mais pertinentes inerentes ao seu desenvolvimento e aplicação surgiram algumas questões que constituem parte dos problemas que carecem de resolução no desenvolvimento de STI. Como forma de contribuir para a possível resolução das questões anteriormente enunciadas, neste trabalho é proposta uma *Framework* de Desenvolvimento de Sistemas Tutores Inteligentes, FD-STI que visa propor técnicas e métodos que se julgam importantes para o desenvolvimento de STI, contribuindo com melhorias para o processo. Após detalhar e aprofundar os componentes da FD-STI, as possíveis contribuições de cada um deles foram explicitadas e referenciadas numa matriz de problemas/contribuições, onde se procura sintetizar e referenciar a importância de cada uma das técnicas, bem como as suas contribuições para uma tentativa de solucionar os problemas identificados.

Também importante, como referido ao longo deste trabalho, é o envolvimento de outros ramos da ciência que não exclusivamente a informática, como as ciências da educação e a psicologia. As teorias de ensino podem ter um papel preponderante nos STI, pois permitem ao sistema socorrer-se de uma teoria de ensino, de entre aquelas que se encontram disponíveis na base de dados de modelos pedagógicos, como proposto na FD-STI, adaptando-se ao aluno e à sua forma de aprender. O aluno deixa de ser responsável por se adaptar ao sistema, o que pode contribuir para melhorias no processo de ensino.

Esta *Framework* é bastante ambiciosa, ao separar completamente o domínio de intervenção do STI, propondo sistemas modulares, capazes de intervir em qualquer domínio de ensino, dado que o conhecimento é armazenado externamente de forma normalizada, bem como as técnicas de ensino, o que no limite permitiria que um sistema acesse a uma base de conhecimento e a uma base de dados de técnicas de ensino, e iniciasse com eficiência o processo de ensinar. Basta comparar esta situação com o facto de um professor ser capaz de ensinar qualquer domínio do conhecimento, para se perceber a ambição da *Framework*.

No âmbito deste trabalho, foi produzido um artigo que foi apresentado na *m-ICTE 2005 – Third International Conference on Multimedia and Information & Communication Technologies in Education*, intitulado “Future Challenges in Intelligent Tutoring Systems – A *Framework*” que se encontra nos 15 artigos mais consultados do site da referida conferência.

A FD-STI é uma mais valia no desenvolvimento de novos sistemas, alertando para os diversos problemas que se levantam e propondo um modelo de referência para o desenvolvimento de Sistemas Tutores Inteligentes.

Como trabalho futuro, pretende-se desenvolver um protótipo, a integrar numa plataforma de e-learning, e.g. Moodle, que é no presente o sistema mais utilizado nas escolas secundárias, aplicando a *Framework* apresentada, no sentido de proporcionar um STI a ser disponibilizado no âmbito do ensino secundário, procurando que possua as características que foram identificadas como desejáveis, obviamente tentando respeitar a independência do sistema relativamente ao domínio, e a possibilidade de adaptação ao aluno e à sua forma de aprender, tarefa que constitui um grande desafio.

Referencias

- [Anderson,84]: Anderson, J. R. (1984) Some reflections on the adquisition of knowledge. Educational Researcher, 5-10.
- [Anderson, 86]: Anderson, J. R., Boyle, C. F., & Yost, G. (1986) The Geometry Tutor. The Journal of Mathematical Behavior, 5-20.
- [Abbas,98]: Abbas, H. Designing a New Domain Knowledge Base for an Intelligent Tutoring System. Ph.D., Illinois Institute of Technology, 1998.
- [AutoTutor]: Auto Tutor: An Artificially Intelligent Tutor, <http://www.autotutor.org>
- [Anderson,83]: Anderson, J. R., The Architecture of Cognition. Cambridge, Massachusetts: Harvard University Press, 1983.
- [Armstrong et al, 95]: Armstrong, R., Freitag, D., Joachims, T., Mitchell, T. WebWatcher: A learning apprentice for the World Wide Web. AAAI Spring Symposium on Information Gathering from Distributed, Heterogeneous Environments, Stanford, CA, 1995.
- [Arruarte et al, 96]: Arruarte, A., Elorriaga J. A., Fernández-Castro, I., Ferrero, B., Knowledge Reusability: Some Experiences in Intelligent Tutoring System. Position paper for ITS'96 workshop on Architectures and Methods for Designing cost-Effective and Reusable ITSs, Montreal, 1996.
- [Ausubel, 68]: Ausubel, D.P., Educational Psychology: A Cognitive View. New York, Holt, Rinehart and Winston, 1968.
- [Beaumont, 94]: Beaumont, I., User Modeling and Hypertext Adaptation in the Tutoring System Anatom-Tutor. Workshop Adaptive Hypertext and Hypermedia at 4-th International Conference on User Modeling, Hyannis, MA, 1994.
- [Bruner, 90]: Bruner, J., Culture and human development: A new look. Human Development, 1990.
- [Burns,Capps, 88]: Burns, H. L. e Capps, C. G., "Foundations of Intelligent Tutoring Systems", em Polson, M. C. e Richardson, J. J. (Ed.), Foundations of Intelligent Tutoring Systems, 21-53, Lawrence Erlbaum Associates, Hillsdale, NJ, 1988

- [Brown et al, 75]: Brown, J., Burton, R. e Bell, A., SOPHIE: a step towards a reactive learning environment. *International Journal of Man-Machine studies*, 1975.
- [Bloom, 84]: Bloom,-Benjamin-S. The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. *Educational-Researcher*; v13 n6 p4-16 Jun-Jul, 1984.
- [Brown, Burton, 78]: Brown, J. S., & Burton, R. R., Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, 2, 155-192, 1978.
- [Brown et al, 82]: Brown, J. S., Burton, R. R., & deKleer, J. Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II and III. In D. Sleeman & J. S. Brown (Eds.), *Intelligent Tutoring Systems* (pp. 227-282). New York: Academic Press, 1982.
- [Burton, 82]: Burton, R. R., Diagnosing bugs in a simple procedural skill. In D. Sleeman & J. S. Brown (Eds.), *Intelligent Tutoring Systems* (pp. 157-183). New York: Academic Press, 1982.
- [Brusilovsky, 96]: Brusilovsky, P., Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6 (2-3), pp. 87-129, 1996.
- [Brusilovsky, 99]: Brusilovsky, P., Adaptive and Intelligent Technologies for Web-based Education. In C. Rollinger and C. Peylo (eds.), *Special Issue on Intelligent Systems and Teleteaching, Künstliche Intelligenz*, 4, 19-25, 1999.
- [Brusilovsky, 01]: Brusilovsky, P., Adaptive hypermedia. *User Modeling and User Adapted Interaction*, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2), 87-110, 2001.
- [Bra, Brusilovsky, 99]: Bra., Paul, Brusilovsky, P., Adaptive Hypermedia: From systems to Framework, *ACM Computing Surveys*, 1999.
- [Bond, 90]: Bond, A, A Computacional Model for Organizations of Cooperating Intelligent Agents, *SIGOIS Bulletin*, 11, 1990.
- [Becker et all, 04]: Becker, K.; Marquardt, C.G.; Ruiz, D.D. A Pre-Processing Tool for Web Usage Mining in the Distance Education Domain. pp. 78-87. 2004.

- [Balabanovic, Shoham, 97]: M. Balabanovic, Y. Shoham. Fab: content-based, collaborative recommendation. In: Commun. ACM, New York, NY, USA, 40(3), páginas 66–72, 1997.
- [Carbonell, 70]: Carbonell J. R.: AI in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction, IEEE Trans. Man-Machine Systems, MMS-11(4), pp 190-202, 1970.
- [Castillo et all, 97]: Castillo, E; Gutiérrez, J.M.;Hadi A.S. Sistemas Expertos y Modelos de Redes Probabilísticas. Monografías de la Academia Española de Ingenieria, Madrid, 1997.
- [Cohen et all, 82]: Cohen, P.A., Kulik, J.A., & Kulik, C.L.C .(1982). Educational outcomes of tutoring: A meta-analysis of findings. American Educational Research Journal, 19, 237-248.
- [Circle]: Circle: Center for Interdisciplinary Research on Constructive Learning Environments, <http://www.pitt.edu/~circle/>
- [Corbett, Anderson, 92]: Corbett, A. T., & Anderson, J. R. (1992). LISP intelligent Tutoring System: Research in Skill Acquisition. In J. H. Larkin & R. W. Chabay (Eds.), Computer-Assisted Instruction and Intelligent Tutoring Systems: Shared Goals and Complementary Approaches (pp. 73-109). Hillsdale, New Jersey: Lawrence Erlbaum Associates, Publishers
- [Circsim] <http://www.cs.iit.edu/~CIRCSIM/>
- [Clancey,Soloway, 90]: Clancey, W. J. e Soloway, E., “Artificial Intelligence and Learning Environments: Preface”, em Clancey, W. J. e Soloway, E. (Ed.), Artificial Intelligence and Learning Environments, 1-6, MIT / Elsevier Science Publishers, Amsterdam, the Netherlands, 1990
- [Carr, Goldstein, 77]: Carr, B. e Goldstein, I., Overlays: a theory of modelling for Computer Aided Instruction, 1977.
- [Cho et all, 00]: Cho, Byung-In, Joel A. Michael, Allen A. Rovick, and Martha W. Evens. An Analysis of Multiple Tutoring Protocols Intelligent Tutoring Systems: 5th International Conference, ITS 2000, Montreal, 2000, Springer, Lecture Notes in Computer Science no. 1839, pp. 212-221.
- [Corbett, Anderson, 92]: Corbett, A. T., & Anderson, J. R. (1992). LISP intelligent tutoring system: Research in skill acquisition. In Larkin, J. H. & Chabay, R. W. (Eds). Computer-assisted instruction and intelligent tutoring systems: Shared goals and complementary approaches. Hillsdale, NJ: Lawrence Erlbaum Associates.02

- [De Rosis et al, 94]: De Rosis, F.; De Carolis, B.; Pizzutilo, S. User-Tailored Hypermedia Explanations. Workshop Adaptive Hypertext and Hypermedia at 4-th International Conference on User Modeling, Hyannis, MA, 1994.
- [Evens et al, 01]: Evens M. W., Stefan B., Ru.Charn C, Freedman , Glass M., Hee Lee Y, Lêem Seop S, Woo Woo C, Yuemei Z, Yujian Z, Joel A.M., Rovick A. A. (2001), CIRCSIM – Tutor: An Intelligent Tutoring System Using Natural Language Dialogues, Twelfth Midwest AI and Cognitive Science Conference. MAICS 2001, Oxford
- [educationaldatamining] <http://www.educationaldatamining.org>
- [El-Sheikh et al, 98]: El-Sheikh, Eman, Sticklen, Jon, (1998), A Framework for Developing Intelligent Tutoring Systems Incorporating Reusability, 11th International Conference on Industrial and Engineering applications of Artificial Intelligence and Expert Systems, Benicassim, Castellon, Spain
- [Freyberger et al, 04]: Freyberger, J.; Heffernan, N.T.; Ruiz, C. Using Association Rules to Guide a Search for Best Fitting Transfer Models of Student Learning. Int. Conf. on Intelligent Tutoring Systems. 2004.
- [Faria, 02]: Faria, Luiz, Treino e Apoio a Operadores de Centros de Controlo e Condução de Redes Eléctricas – Uma Abordagem Baseada em Conhecimento e Tutores Inteligentes, tese de Doutoramento, FEUP Universidade do Porto, 2002
- [Fernandez, 89]: Fernandez I. 1989, Estratégias de Enseñanza en un sistema Inteligente de Enseñanza Assistida por Ordenador, Tesis Doctoral de la Universidad del País Vasco, San Sebastian
- [Felder, Soloman, 88]: Felder, R, Soloman, B (1988) Learning Styles and Strategies <http://www.ncsu.edu/felder-public/ILSdir/styles.htm>
- [Fisher et al, 90]: Fischer, G.; Mastaglio, T.; Reeves, B. Minimalist Explanations in Knowledgebased Systems. 23th Annual Hawaii International Conference on System Sciences, Kailua-Kona, HI. p. 309-317, 1990.
- [Fisher, 98]: Fischer, G. Beyond Couch Potatoes: from consumers to designers. In Proceedings of the 5th Asia Pacific Computer-Human Interaction Conference. IEEE Computer Society, 2-9, 1998.

- [Fisher, 01]: Fischer, G User modeling in human-computer interaction. User Modeling and User-Adapted Interaction 11(1-4), 65-86, 2001.
- [Fowler91]: Fowler, D. G. A Model for Designing Intelligent Tutoring Systems. Journal of Medical, 1991
- [Frasson, 97]: Frasson, C. et al. Using Pedagogical Agents in a Multi-Strategic Intelligent Tutoring System. In: World Conference On Artificial Intelligence Tutoring Agents, 8., 1997, Kobe: Springer-Verlag, (1997).
- [Freeman, 00]: Freeman, R. What is an Intelligent Tutoring System ? Published in Intelligence, 11(3):15-16, 2000.
- [Frigo et al, 04]: Frigo, L.B., Pozzebon E., Bittencourt, G., O Papel dos Agentes Inteligentes nos Sistemas Tutores Inteligentes, Proceedings of the World Congress on Engineering and Technology Education, 2004, SP
- [Fink, Kobsa, 00]: Fink, J., Kobsa, A. A review and analysis for commercial user modeling servers for personalization on the World Wide Web. User Modeling and User Adapted Interaction 10, 209-249, 2000.
- [Frasson et all, 98]: Frasson, Claude; Martin, Louis; Gouarderes, Guy; Aimeur, Esma. LANCA: A Distance Learning Architecture Based on Networked cognitive Agents. In Lectures Notes in Computer Science. Intelligent Tutoring Systems. Proceedings of 4th International Conference, ITS 1998, San Antonio, Texas, August 1998. P.594- 603.
- [Giraffa, Viccari, 99]: Giraffa, L. M. M.; Viccari, R. M. Intelligent tutoring systems built using agents techniques. Revista de Educação, Ciência e Cultura, Canoas: La Salle, p.23–40, 1999.
- [Gouarderes et all, 99]: Gouarderes, G.; Françoise Canut, C. M.; Sanchis, E. The SYSTEMION: a new agent model to design intelligent tutoring system. IOS Press, 1999. p.54–63.
- [Gruber, 93]: Gruber, T, (1993). A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2), 199-220.
- [Gambôa, Fred, 01]: Gambôa, H., Fred, A., Designing Intelligent Tutoring Systems: a Bayesian Approach., 3rd International Conference on Enterprise Information Systems, ICEIS'2001, 2001.
- [Gomes et all, 05]: Gomes, P. and Antunes, B. and Rodrigues, L. and Santos, A. and Barbeira, J. and Carvalho, R. , "Using Ontologies for eLearning Personalization", 3rd E-learning Conference – Computer Science Education, Coimbra, Portugal, September

- [Goldberg et al, 92]: D. Goldberg, D., D. Nichols, B. M. Oki, D. Terry. Using collaborative filtering to weave an information tapestry. In Commun. ACM, New York, NY, USA, 35(12), páginas 61–70, 992.
- [Gagné et all, 92]: Gagné, R. M., Briggs, L. e Wager, W., “Principles of Instructional Design”, Harcourt Brace Jovanovich, Orlando, FL, 1992.
- [Guardia, 93]: Guardia Robles, B. (1993) Asesores Inteligentes para apoyar el processo de enseñanza de lenguages de programacion. Tesis de Grado. Asesor. ITESM: Instituto Tecnológico de Monterrey
- [Gardner, Hatch, 89]: Gardner, H., & Hatch, T. (1989). Multiple intelligences go to school: Educational implications of the theory of multiple intelligences. Educational Researcher, 18(8), 4-9.
- [Ha et all, 00]: Ha, S.H.; Bae, S.M.; Park, S.C. Web mining for distance education. APAN Conference. Beijing, 2000.
- [Hollan et all, 81]: James D. Hollan, Michael D. Williams, and Al Stevens. An overview of Steamer. Behavior Research Methods and Instrumentation, 13:85-90, 1981. Hollan James D.,An overview of Steamer, June 1985. Invited talk given to Hudson Valley Special Interest Group for Artificial Intelligence; Vassar College, NY.
- [Hewitt et all, 73]: Carl Hewitt, Peter Bishop and Richard Steiger. A Universal Modular Actor Formalism for Artificial Intelligence IJCAI. 1973.
- [Hook et all, 96]: Höök, H., Böcker, H., Gunzenhauser, R. A Glass Box Approach To Adaptive Hypermedia. User Models And User Adapted Interaction. v. 6, 1996.
- [Iglesias et all, 01]: Iglesias C.A.,Garijo C., Gonzalez J. (2001), Metodologias Orientadas a Agentes, Inteligencia artificial, Revista Iberoamericana de Inteligencia artificial. Nº 6 Vol. 2
- [Jameson et all, 99]: Jameson, A., Schäfer, R., Weis, T., Berthold, A., & Weyrath, T. (1999) Making Systems Sensitive to the User’s Changing Resource Limitations Knowledge-Based Systems, 12, 413–425.

- [Jonassen, 1984]: Jonassen, D., Wang, S. Acquiring structural knowledge from semantically structured hypertext. Journal of Computer-based Instruction. Vol. 20, Nº 1, p. 1-8,1993.
- [Jonassen, 88]: Jonassen, D. H. Designing Structured Hypertext and Structuring Access to Hypertext, Educational Technology, pp.13-16, November 1988.
- [Jonassen, Wang, 93]: Jonassen, D., Wang, S. Acquiring structural knowledge from semantically structured hypertext. Journal of Computer-based Instruction. Vol. 20, Nº 1, p. 1-8, 1993.
- [Johnson, 1984]: Johnson, W. L. and Soloway, E. (1984). Proust: Knowledge-based program understanding. In In Proceedings of the 7th international conference on Software engineering, Florida, United States
- [Jonhson, 86]: Johnson, W. L.,Intention-Based Diagnosis of Novice Programming Errors, Morgan-Kauffman (1986)
- [Jennings et all, 96]: Jennings, N. R.; Faratin, P.; Johnson, M. J.; O'brien, P.; Wiegand, M. E. Using Intelligent Agents to Manage Business Processes, Proceedings of First International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM96), London, UK, 345-360.
- [Kaplan et all, 93]: Kaplan, C.; Fenwick, J.; Chen, J. Adaptive Hypertext Navigation Based On User Goals And Context. User Models And User-Adapted Interaction. V. 3, N. 3, P.193-220, 1993.
- [Kay, Kummerfeld, 94]: Kay, J.; Kummerfeld, R. J. An Individualised Course for the C Programming Language Second International WWW Conference "Mosaic and the Web", Chicago,IL, 1994.
- [Kintsch, 93]: Kintsch, E., "Principles of Instruction from Research on Human Cognition", em Spector, J. M., Polson, M. C. e Muraida, D. J. (Ed.), Automating Instructional, 1993
- [Kinshuk, Patel, 97]: Kinshuk Ashok Patel, (1997),A Conceptual Framework for Internet based Intelligent Tutoring Systems, Knowledge Transfer (Volume II) (Ed. A. Behrooz), 1997, pAce, London, UK, pp117-124 (ISBN -900427-015-X)
- [Karlsson, 95]: Karlsson, E. A., Software Reuse – A Holistic Approach, Wiley Series in Software Based systems, 1995
- [Kinshuk, Patel, 96] : Kinshuk & Patel A. (1996). Knowledge Characteristics: Reconsidering the design of Intelligent Tutoring Systems.

- Knowledge Transfer (Ed. Behrooz A.), pp190-197, ISBN 1 900427 01 X
- [Kolodner, 88]: Kolodner, J., Proceedings of the first case-based reasoning workshop. Morgan Kaufmann, 1988.
- [Kinshuk, 02]: Kinshuk. Does intelligent tutoring have future! Proceedings of the Inter. Conf. on Computers in Education, Los Alamitos, CA: IEEE Computer Society (2002).
- [Kobsa, 01]: Kobsa, A. Generic user modeling systems. User Modeling and User-Adapted Interaction 11(1-4), 49-63, 2001
- [Khuwaja, 94]: Khuwaja, R.A. (1994) A Model of Tutoring: Facilitating Knowledge Integration Using Multiple Models of the Domain. Dissertação de Doutorado, Illinois Institute of Technology
- [Li, Zaiane, 04]: Li, J.; Zaiane, O.R. Combining Usage, Content and Structure Data to Improve Web Site Recommendation. Int. Conf. on Electronic Commerce and Web Technologies. Spain. 2004.
- [Lelouche et al, 03]: Lelouche, Ruddy, Ly, Tho Toan, (2003), Using a Framework in the Development of an Intelligent Tutoring System, IEEE International Conference on Information Reuse and Integration, IRI 2003.
- [Liegle, Woo, 00]: Liegle J., Woo H., Developing Adaptive Intelligent tutoring Systems: A framework and its implementations, Presentation and publication in the proceedings of the ISECON Conference, Philadelphia, November 9-12 (2000).
- [Mathé, Chen, 94]: Mathé, N., Chen, J. A User-Centered Approach to Adaptive Hypertext based on an Information Relevance Model. 4-th International Conference on User Modeling, Hyannis, MA. p.107-114, 1994.
- [Marcke, Sarti, 95]: Marcke, K. Van, Sarti, L. - Reuse in Intelligent Courseware Authoring (August 1995) - AI-ED-95 Workshop on Authoring Shells for Intelligent Tutoring Systems
- [Mctaggart, 01]: Mctaggart, J. Intelligent Tutoring System and Education for the Future. Literature Review, 2001.
- [Mizoguchi, Bourdeau, 00]: Mizoguchi R., Bourdeau J.,(2000, Using Ontological Engineering to Overcome Common AI-ED Problems, International Journal of Artificial Intelligence in Education 11, 1-12.

- [Maes, 90]: Maes, P. (1990). Situated Agents Can Have Goals. In Maes, P., editor, *Designing Autonomous Agents*. MIT Press.
- [Mizukami, Nicoletti, 86]: Mizukami, Maria da Graça Nicoletti., *Ensino: as abordagens o processo*, São Paulo:EPU, 1986.
- [Murray et all, 01]: Murray, T., Gartner-Piemonte, J., & Kelleher, P. (2001). Evaluating Features for Conceptual and Narrative Flow in an Adaptive Hyperbook. *Proceedings of the International Conference on Artificial Intelligence and Education: AIED-2001*, 21-32.
- [Novak, Gowin, 84]: Novak, J. D. & Gowin, D., 1984. *Learning How to Learn*. Cambridge: Cambridge University Press.
- [Nielsen, 93]: Nielsen, J. *Usability Engineering*. Academic Press. 1993.
- [Novais, 03]: Novais, Paulo (2003), *Teoria dos Processos de Pré-Negociação em Ambientes de Comércio Electrónico*, Tese de Doutoramento em Informática, Universidade do Minho
- [Novais et all, 06]: Novais Paulo, Henriques Pedro, Lopes Hugo, Martins Ricardo, *Issues in Personalizing e-Commerce System*, Universidade do Minho
- [Negoita, Reusch, 04]: Negoita M., Reusch B, *Basics of engineering the intelligent hybrid systems, The eight international conference on knowledge-based intelligent information & engineering systems KES'2004*.
- [O'Shea, 82]: O'Shea, T.. A self-improving quadratic tutor. In D. Sleeman & J. S. Brown (Eds.), *Intelligent Tutoring Systems* (pp. 309-336). New York: Academic Press, 1982
- [Piaget, 89]: Piaget, J. (1989) *Les relations entre l'intelligence et l'affectivité dans le developpement de l'enfant*. Rimé, B.; Scherer, K. (Eds.) *Les Émotions. Textes de base en psychologie*. Paris: Delachaux et Niestlé, 1989.
- [Perkowitz, Etzioni, 98]: Perkowitz, M.; Etzioni, O. *Adaptive web sites: Automatically synthesizing web pages*. National Conference on Artificial Intelligence. WI. 1998.
- [Prates, De Souza, 00]: Prates, R. O.; De Souza, C. S. *A Case Study For Evaluating Interface Design Through Communicability*. *Proceedings of the*

- International Conference on Designing Interactive Systems, DIS2000, New York, NY: ACM Press, 308-317, 2000.
- [Perkins, 95]: Perkins, D. (1995) La escuela inteligente. Gedisa. Johnson-Laird,
- [Pressman, 95]: Pressman, R.S., Engenharia de Software, McGraw-Hill, 1995
- [Pahl, 04]: Pahl, C. Data Mining Technology for the Evaluation of Learning Content Interaction. International Journal on E-Learning IJEL. 2004.
- [Russell, Norvig, 03]: Russell, S.J. & Norvig P., 2003, Artificial Intelligence: A Modern Approach (2nd edition, Prentice Hall
- [Rovick, Brenner, 83]: Rovick A., Brenner L, 1983, HEARTSIM: a cardiovascular Simulation with Didactic Feedback, The Physiologist, vol. 26 nº 4
- [Rao, George, 95]: Rao, A. and George, P. (1995). BDI Agents: from Theory to Practice. In Proceedings of the First International Conference on Multi-Agent Systems - ICMAS95, San Francisco, USA.
- [Rosatelli et al, 00]: Rosatelli, M.C., Self, J.A. and Thiry, M. LeCS: a collaborative case study system, in G. Gauthier, C. Frasson and K. VanLehn (eds.), Intelligent Tutoring Systems, Berlin: Springer-Verlag, p232-24, 2000.
- [Rodrigues et al, 05]: Rodrigues, Manuel, Novais, Paulo, Santos, M. Filipe, (2005), Future Challenges in Intelligent Tutoring Systems – A Framework, m-ICTE 2005 – Tercera Conferencia Internacional sobre Multimedia y Tecnologías de la Informacion e Comunicación en Educación – Cáceres – España
- [Rovick, Michael, 86]: Rovick A., Michael J. 1986, CIRCSIM: IBM-PC Computer Teaching Exercise on Blood Pressure Regulation, Proceedings XXX congress of International Union of Physiological sciences, Vancouver , Canada
- [Rowley, 96]: Rowley, K., Increasing the Effectiveness of ITS Research and Development, ITS'96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs, Montreal 1996
- [Riesbeck, Schank, 91]: Riesbeck, C. e Schank, R., From training to teaching: techniques for case-based ITS. Em Intelligent Tutoring systems: evolutions in design, Laurence Erlbaum Associates, 1991.

- [Reategui, Lorenzatti, 05]: E. Reategui, A. Lorenzatti. Um Assistente Virtual para Resolução de Dúvidas e Recomendação de Conteúdo. In: ENIA 2005, São Leopoldo, RS. páginas 821-830, 2005.
- [Santos et all, 06]: Santos, A. and Gomes, P. and Antunes, B. and Rodrigues, L. and Barbeira, J. , "PERSONA: Personalização de Serviços com base na Modelização Inteligente de Utilizadores", 1ª Conferência Ibérica de Sistemas e Tecnologias de Informação, Esposende, Portugal, June 2006
- [Self, 99]: Self, John. The defining characteristics of intelligent tutoring systems research: itss care, precisely. International Journal of Artificial of Artificial Intelligence in Education, Leeds, England, 1999.
- [Schank, 82]_: Schank, R., Dynamic memory: a theory of learning in computers and people, Cambridge University Press, 1982.
- [Sierra et all, 03]: Sierra, E.; Hossian, A. y García-Martínez, R. (2003). Sistemas Expertos que recomiendan estrategias de instrucción. Un Modelo para su Desarrollo. Revista Latinoamericana« de Tecnología Educativa. Volumen 1 N° 1. Páginas 19-30. Facultad de Educación. Universidad de Extremadura. ISSN: 1695-288X.
- [Silva, 98]: Silva, António, (1998) Tutores Inteligentes para Treino de Operadores de Centros de Controlo e Condução de Redes Eléctricas – Sistema SPARSE-IT, Tese de Mestrado em Engenharia Electrotécnica e Computadores, Faculdade de Engenharia da Universidade do Porto.
- [Silva et all, 01]: Silva, D. R.; Seno, W. P.; Vieira, M. T. P. (2001). Acompanhamento do Aprendizado em Educação a Distância com Uso de Data Mining. In: Conferência Latinoamericana de Informática, Mérida, Venezuela, 2001.
- [Skinner, 58]: Skinner, B.F., Teaching Machines. Science 128, pages 969-977, 1958
- [Schafer et all, 01]: Schafer, J.B., Konstan, J. A. and Riedl, J. January 2001. ECommerce Recommendation Applications. Journal of Data Mining and Knowledge Discovery.
- [Srivastava et all, 00]: Srivastava, J.; Mobasher, B.; Cooley, R. Automatic Personalization Based on Web Usage Mining. Communications of the Association of Computing Machinery. pp. 142-151. 2000
- [Sleeman, Hartley, 73]: Sleeman & Hartley, J, D. Towards more intelligent teaching systems. International Journal of Man-Machine Studies (1973). 2, 215-236.0

- [Self, 74]: Self, J.A.. Student models in computer-aided instruction, International Journal of Man.-machine Studies (1974), 6, 261-276
- [Self, 1999]: Self, John, The defining characteristics of intelligent tutoring systems research: its care, precisely. International Journal of Artificial of Artificial Intelligence in Education, Leeds, England, 1999.
- [Santos, 03]: Santos, M.Filipe, Azevedo, C., Data Mining – Descoberta de Conhecimento em Bases de Dados, FCA – Editora de Informática
- [Sleeman, Brown, 82]: Sleeman, D., & Brown, J. S.. Introduction: Intelligent Tutoring Systems. In D. Sleeman & J. S. Brown (Eds.), Intelligent Tutoring Systems (1982) (pp. 1-11). New York: Academic Press.
- [Stevens, Collins, 77]: Stevens A.,Collins A. , The Goal Structure of a Socratic Tutor, In Proceedings of the National ACM Conference, New Work: ACM
- [Selker, 89] : Selker, T, 1989, Cognitive Adaptative Computer Help (Coach), 4TH Intl. Conference on AI and Education, Elsevier Science Publishers
- [Thibodeau et all, 00]: Marc-André Thibodeau, Simon Bélanger, Claude Frasson: WHITE RABBIT - Matchmaking of User Profiles Based on Discussion Analysis Using Intelligent Agents. Intelligent Tutoring Systems 2000: 113-122
- [Tan, 99]: Tan, A.-H. (1999). "Text mining: The state of the art and the challenges". In Proceddings, PAKDD'99 workshop on Knowledge Discovery from Advanced Databases, Beijing, pp.65-70
- [Villareal, Giraffa, 01]: Villareal Goulart R. R.;Giraffa M.L. 2001 Arquiteturas de Sistemas Tutores Inteligentes
- [Vanlehn, 88]: Vanlehn, K. Student Modelling. In Polson, M. C. and Richardson, J. J., editors,Foundations of Intelligent Tutoring Systems, chapter 3, pages 55 - 78. Lawrence Erlbaum Associates Publishers, New Jersey, 1988.
- [Vygotsky, 78]: Vygotsky, L.S. (1978) Mind in Society. Cambridge, MA: Harvard University Press, 1978.

- [Villareal, Giraffa, 01]: Villarreal Goulart, R. Y Giraffa, M. L. (2001). Utilizando a Tecnologia de Agentes na Construção de Sistemas Tutores Inteligentes em Ambientes Interactivos. Tese de Mestrado. PUCRS
- [VanLehn et all, 05]: VanLehn, K., Lynch, C., Schulze, K., Shapiro, J.A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., and Wintersgill, M. (2005). The Andes Physics Tutoring System: Lessons Learned. International Journal of Artificial Intelligence and Education, 15 (3).
- [Vassileva et all, 01]: Vassileva, Julita; Deters, Ralph; Geer, Jim; Mccalla, Gord; BULL, Susan; Kettel, Lori. Lessons from Deploying I-Help. Workshop – Multi-Agent Architectures for Distributed Learning Environments. Proceedings International Conference on AI and Education, San Antonio, Texas, May, 2001. .3-11.
- [Vassileva et all, 01]: Vassileva, Julita; Deters, Ralph; Geer, Jim; Mccalla, Gord; BULL, Susan; Kettel, Lori. Lessons from Deploying I-Help. Workshop – Multi-Agent Architectures for Distributed Learning Environments. Proceedings International Conference on AI and Education, San Antonio, Texas, May, 2001. P.3-11.
- [Wooldridge, 99]: Wooldridge, M. (1999). Intelligent Agents. In Weiss, G., editor, Multiagent Systems - A modern Approach to Distributed Artificial Intelligence. MIT Press.
- [Wooldridge, Jennings, 95]: Wooldridge, M. and Jennings, N. (1995). Intelligents agents: Theory and Practice. The Knowledge Engineering Review, 10(2):115-152.
- [Wenger, 87]: Wenger, E. Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communications of Knowledge. Los Altos, CA: Morgan Kaufmann Publishers, 1987.
- [Wolf, 84]: Wolf, B. 1984 Context Dependent Planning in a Machine Tutor, Phd. Dissertation, University of Massachussets, Amherst, Massachussets
- [Wang et all, 04]: Wang, W; Weng, J.; Su, J.; Tseng, S. Learning Porfolio Analysis and Mining in SCORM Compliant Environment. IEEE Frontiers in Education Conference. 2004.
- [Webber et all, 01]: Webber, Carine; Bergia, Loris; Pesty, Sylvie; Balacheff, Nicolas. The Baghera project: a multi-agent architecture for

human learning. Workshop – Multi-Agent Architectures for Distributed Learning Environments. Proceedings International Conference on AI and Education, San Antonio, Texas, May, 2001. P.12-17.

- [Yang et al., 02]: Yang A., Kinshuk & Patel A. (2002), A Plug-able Web-based Intelligent Tutoring Systems, in S. Wrycza (ed.) Proceedings of the Xth european conference on Information Systems, Gdansk, Poland.
- [Zouaq et al., 00]: Zouaq, Amal; Frasson, Claude; Rouane, Khalid. The Explanation Agent. In Lectures Notes in Computer Science. Intelligent Tutoring Systems. Proceedings of 5th International Conference, ITS 2000, Montreal, Canada, June 2000. P.554-563.
- [Zytkow, Klosgen, 01]: Zytkow J.; Klosgen W., Handbook of Data Mining and Knowledge Discovery. Oxford University Press. 2001.
- [Zaiane, 01]: Zaiane, O.Z. Web Usage Mining for a Better Web-Based Learning Environment. Conference on Advanced Technology for Education. pp 60-64. Alberta. 2001.
- [Zaiane, 02]: Zaiane, O.R. Building a Recommender Agent for e-Learning Systems. International Conference on Computers in Education. New Zealand. pp 55-59. 2002.